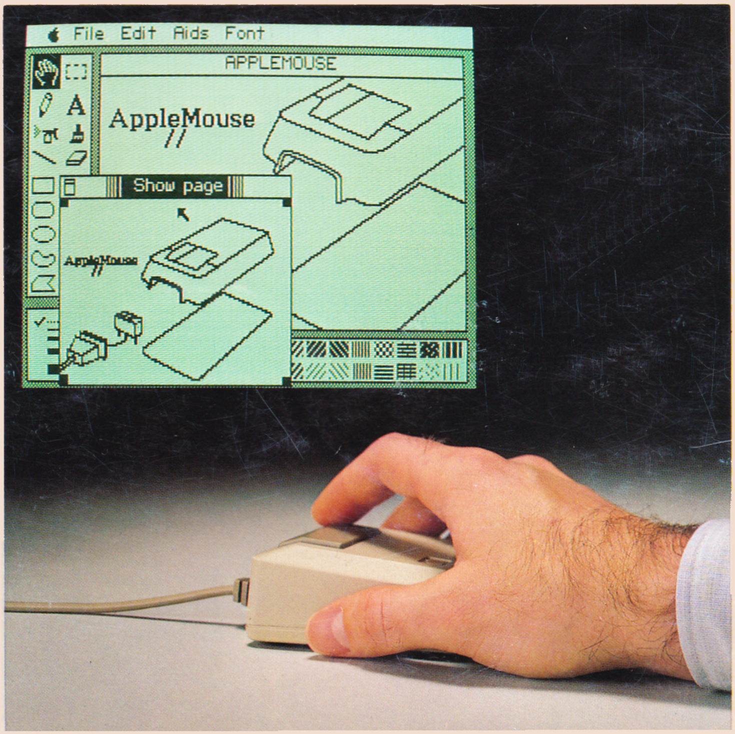
AppleMouse *II* User's Manual

For the Apple *II*e, II Plus, and II


File Edit Aids Font
APPLEMOUSE
AppleMouse //
Show page

Customer Satisfaction

If you discover physical defects in the manuals distributed with an Apple product or in the media on which a software product is distributed, Apple will replace the documentation or media at no charge to you during the 90-day period after you purchased the product.

In addition, if Apple releases a corrective update to a software product during the 90-day period after you purchased the software, Apple will replace the applicable diskettes and documentation with the revised version at no charge to you during the six months after the date of purchase.

In some countries the replacement period may be different; check with your authorized Apple dealer. Return any item to be replaced with proof of purchase to Apple or an authorized Apple dealer.

Limitation on Warranties and Liability

Even though Apple has tested the software described in this manual and reviewed its contents, neither Apple nor its software suppliers make any warranty or representation, either express or implied, with respect to this manual or to the software described in this manual, their quality, performance, merchantability, or fitness for any particular purpose. As a result, this software and manual are sold "as is," and you the purchaser are assuming the entire risk as to their quality and performance. In no event will Apple or its software suppliers be liable for direct, indirect, incidental, or consequential damages resulting from any defect in the software or manual, even if they have been advised of the possibility of such damages. In particular, they shall no have no liability for any programs or data stored in or used with Apple products, including the costs of recovering or reproducing these programs or data. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Copyright

This manual and the software (computer programs) described in it are copyrighted by Apple or by Apple's software suppliers, with all rights reserved. Under the copyright laws, this manual or the programs may not be copied, in whole or part, without the written consent of Apple, except in the normal use of the software or to make a backup copy. This exception does not allow copies to be made for others, whether or not sold. but all of the material purchased (with all backup copies) may be sold, given or loaned to another person. Under the law, copying includes translating into another language.

You may use the software on any computer owned by you but extra copies cannot be made for this purpose. For some products, a multi-use license may be purchased to allow the software to be used on more than one computer owned by the purchaser. including a shared-disk system. (Contact your authorized Apple dealer for information on multi-use licenses.)

Product Revisions

Apple cannot guarantee that you will receive notice of a revision to the software described in this manual, even if you have returned a registration card received with the product. You should periodically check with your authorized Apple dealer.

©1983 Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, California 95014

Apple, the Apple logo, ImageWriter, Silentype, and MousePaint are registered trademarks of Apple Computer, Inc.

Simultaneously published in the U.S.A. and Canada. All rights reserved.

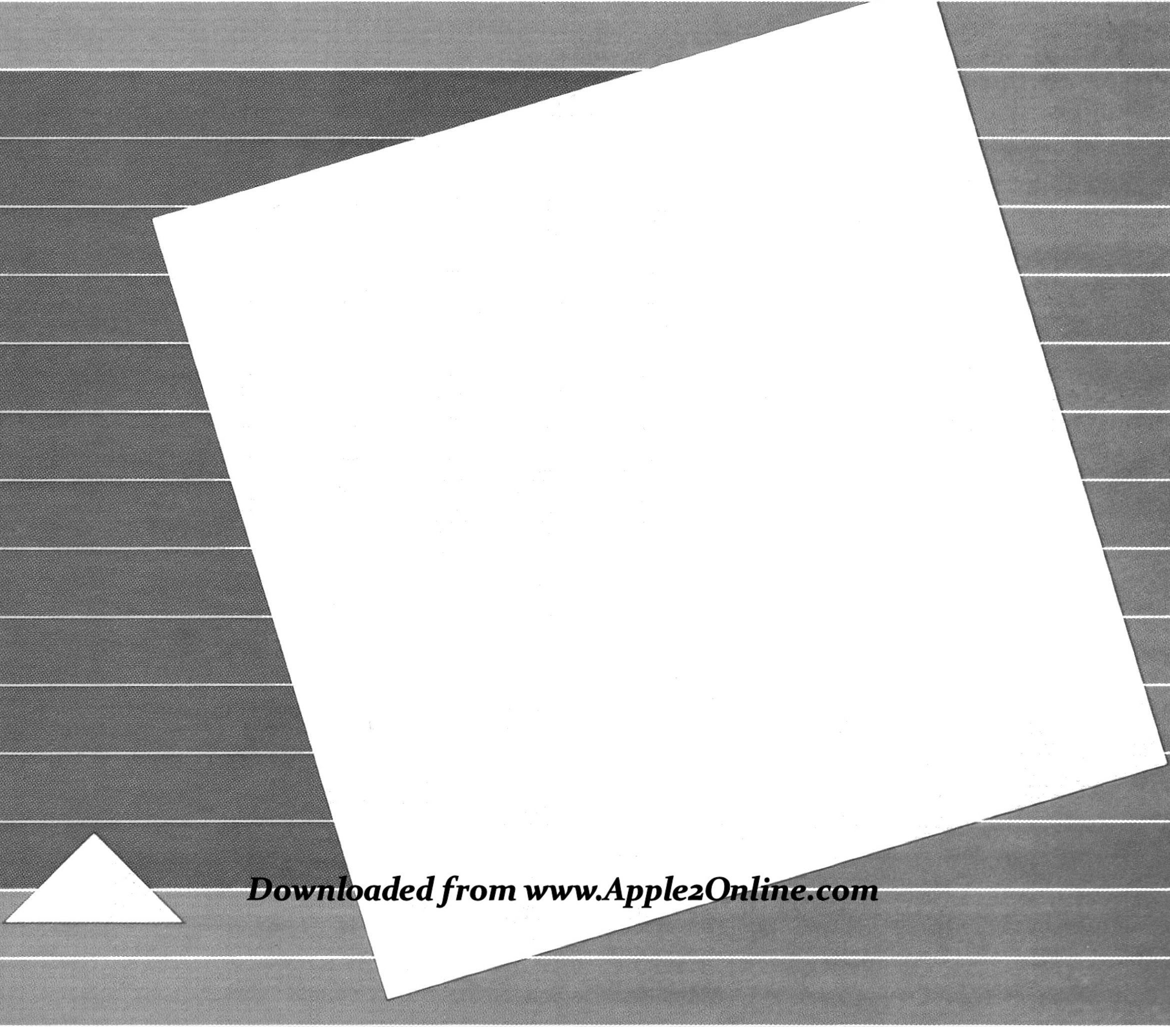
Warning

This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

Reorder Apple Product #A2L2068

Apple II

AppleMouse II User's Manual



Downloaded from www.Apple2Online.com

Radio and Television Interference

The equipment described in this manual generates and uses radio-frequency energy. If it is not installed and used properly, that is, in strict accordance with our instructions, it may cause interference with radio and television reception.

This equipment has been tested and complies with the limits for a Class B computing device in accordance with the specifications in Subpart J, Part 15, of FCC rules. These rules are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that the interference will not occur in a particular installation, especially if you use a "rabbit ear" television antenna. (A "rabbit ear" antenna is the telescoping-rod type usually contained on TV receivers.)

You can determine whether your computer is causing interference by turning it off. If the interference stops, it was probably caused by the computer or its peripheral devices. To further isolate the problem:

- Disconnect the peripheral devices and their input/output cables one at a time. If the interference stops, it is caused by either the peripheral device or its I/O cable. These devices usually require shielded I/O cables. For Apple peripheral devices, you can obtain the proper shielded cable from your dealer. For non-Apple peripheral devices, contact the manufacturer or dealer for assistance.

If your computer does cause interference to radio or television reception, you can try to correct the interference by using one or more of the following measures:

- Turn the TV or radio antenna until the interference stops.
- Move the computer to one side or the other of the TV or radio.
- Move the computer farther away from the TV or radio.
- Plug the computer into an outlet that is on a different circuit than the TV or radio. (That is, make certain the computer and the radio or television set are on circuits controlled by different circuit breakers or fuses.)
- Consider installing a rooftop television antenna with coaxial cable lead-in between the antenna and TV.

If necessary, you should consult your dealer or an experienced radio/television technician for additional suggestions. You may find helpful the following booklet, prepared by the Federal Communications Commission:

"How to Identify and Resolve Radio-TV Interference Problems."

This booklet is available from the U.S. Government Printing Office, Washington, DC 20402, stock number 004-000-00345-4.

Table of Contents

List of Figures vii

About This Manual ix

1 **Here Comes the Mouse** 1

2 **Hooking It Up** 3

3 Installing the New Connector
4 Apple IIe Installation
6 Apple II or Apple II Plus Installation
8 Installing the AppleMouse II Card
10 Connecting the Mouse
10 Your Mouse's Working Area

3 **MousePaint** 13

13 Getting Started
14 The MousePaint Display
15 Making a Picture
15 Getting a Blank Screen
15 Selecting Line Width and Pattern
16 Drawing a Shape
16 Using the Brush
16 Modifying Your Picture
17 Drawing Tools
17 Grabbing Hand
17 Editor's Box
17 Pencil
18 Text Letter

18 Spray Can
18 Brush
19 Straight Edge
19 Eraser
19 Enclosed Shapes
20 Editing Commands
20 Move
21 Cut and Paste
21 Copy
22 Delete
22 Invert
22 Flip Horizontally and Vertically
22 Undo
23 Drawing Aids Commands
23 FatBits
24 Show Page
24 Set Brush
24 Grid
24 Filing Commands
24 Blank Screens
25 Disk Names and Filenames
25 Acceptable Names
25 Calling a File
25 Examples
26 Formatting a Storage Disk
26 Put a Copy In
27 Get Picture
27 Put Back Picture
28 Delete File
28 Print Picture

4 ***Care and Feeding*** ***31***

A ***Writing Mouse Programs in BASIC*** ***35***

35 The BASIC Mouse
35 Calling Your Mouse
36 Reading the Mouse Position
37 Turning off the Mouse
37 A Sample Program: MOUSE.MOVE
37 Comments
38 A Sample Program: MOUSE.DRAW
39 Comments

B

AppleMouse II Firmware 43

43 Finding the Mouse
44 Setting the Operating Mode
44 Reading Mouse Data
45 Passive Mode
46 Interrupt Modes
46 Mouse Routines
47   SETMOUSE
47   SERVEMOUSE
47   READMOUSE
47   CLEARMOUSE
47   POSMOUSE
48   CLAMPMOUSE
48   HOMEMOUSE
48   INITMOUSE
49 Calling the Mouse Routines
50 A Sample Program

Downloaded from www.Apple2Online.com

List of Figures

Chapter 2: Hooking It Up

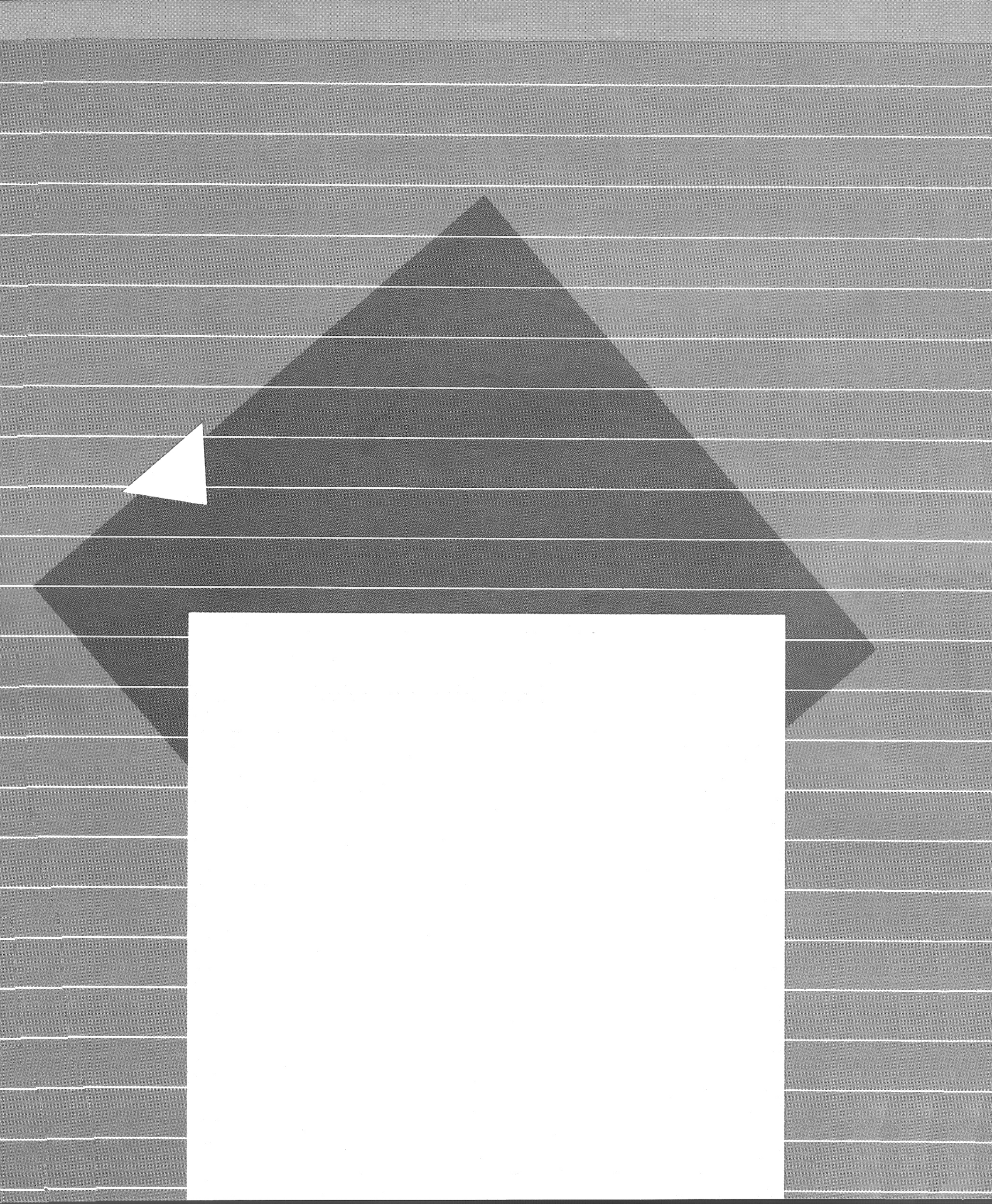
4 Figure 2-1. Apple IIe Connector Parts
4 Figure 2-2. Threading a Jack Screw Into the Connector
5 Figure 2-3. Placing the Connector in the Panel Opening (Back View)
5 Figure 2-4. Tightening the Jack Screws
6 Figure 2-5. Apple II or II Plus Connector Parts
6 Figure 2-6. Attaching the Cable to the Cable Clamp
7 Figure 2-7. Attaching the Connector to the Clamp
7 Figure 2-8. Assembling the Clamp
8 Figure 2-9. Installing the Clamp on the Back Panel
9 Figure 2-10. Plugging in the Card

Chapter 3: MousePaint

14 Figure 3-1. The MousePaint Display

Chapter 4: Care and Feeding

31 Figure 4-1. The Underside of Your Mouse



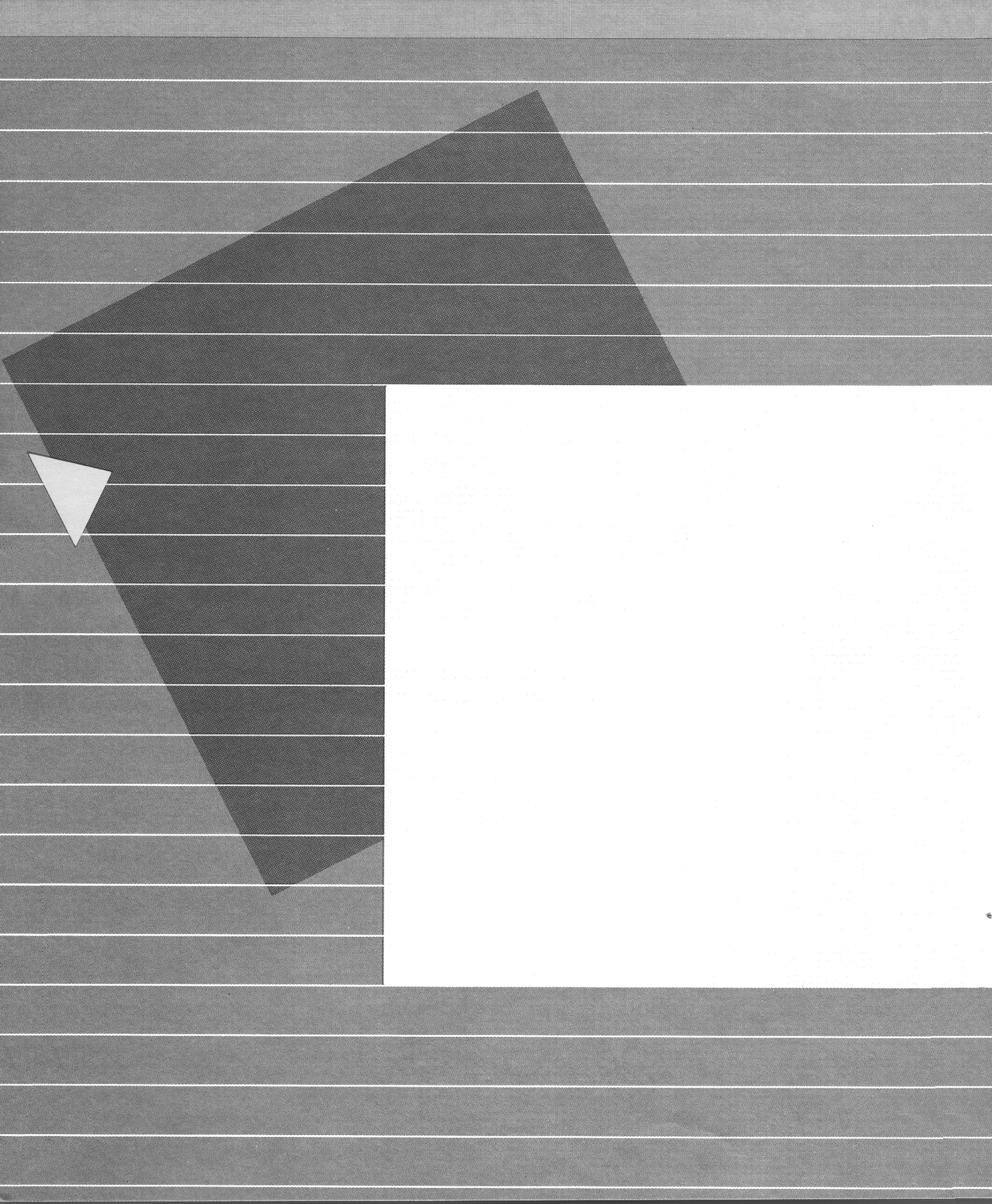
About This Manual

This manual tells you all about installing and using your AppleMouse II. Here is a summary:

- Chapter 1 introduces you to the mouse and tells you some of the things it can do.
- Chapter 2 presents detailed instructions for hooking your mouse up to your Apple IIe, Apple II, or Apple II Plus computer.
- Chapter 3 explains MousePaint, a fascinating graphics program included in your mouse package.
- Chapter 4 tells you how to take care of your mouse.

You can hook up and enjoy your mouse without knowing anything about programming. However, if you are a programmer, you will find the appendixes useful:

- Appendix A contains instructions and sample routines if you want to write your own mouse programs in BASIC.
- Appendix B is for experienced machine-language programmers. It contains a technical description of the 6502 assembly language interface between the AppleMouse II and your Apple computer.



Here Comes the Mouse

The AppleMouse II is a hand-held device, connected to your Apple computer by a flexible cable. As you roll your mouse about on the table next to your computer, it makes a cursor move in exactly the same way on your monitor screen. By moving the cursor to different symbols on the screen, you can tell your Apple what you want it to do. The AppleMouse II represents one of the simplest and most versatile computer input devices available today.

Mouse Programs

Your mouse requires special software. It works only when you have loaded your computer with a program that has been written for it. One such program, MousePaint, is on the disk that came with your mouse. Other AppleMouse II programs are being issued constantly; check with your Apple dealer for details. If you understand programming, you can even write your own mouse programs; the appendixes in this manual will get you started.

Caution: When you buy programs to use with your mouse, make sure that they are designed to be compatible with the AppleMouse II.

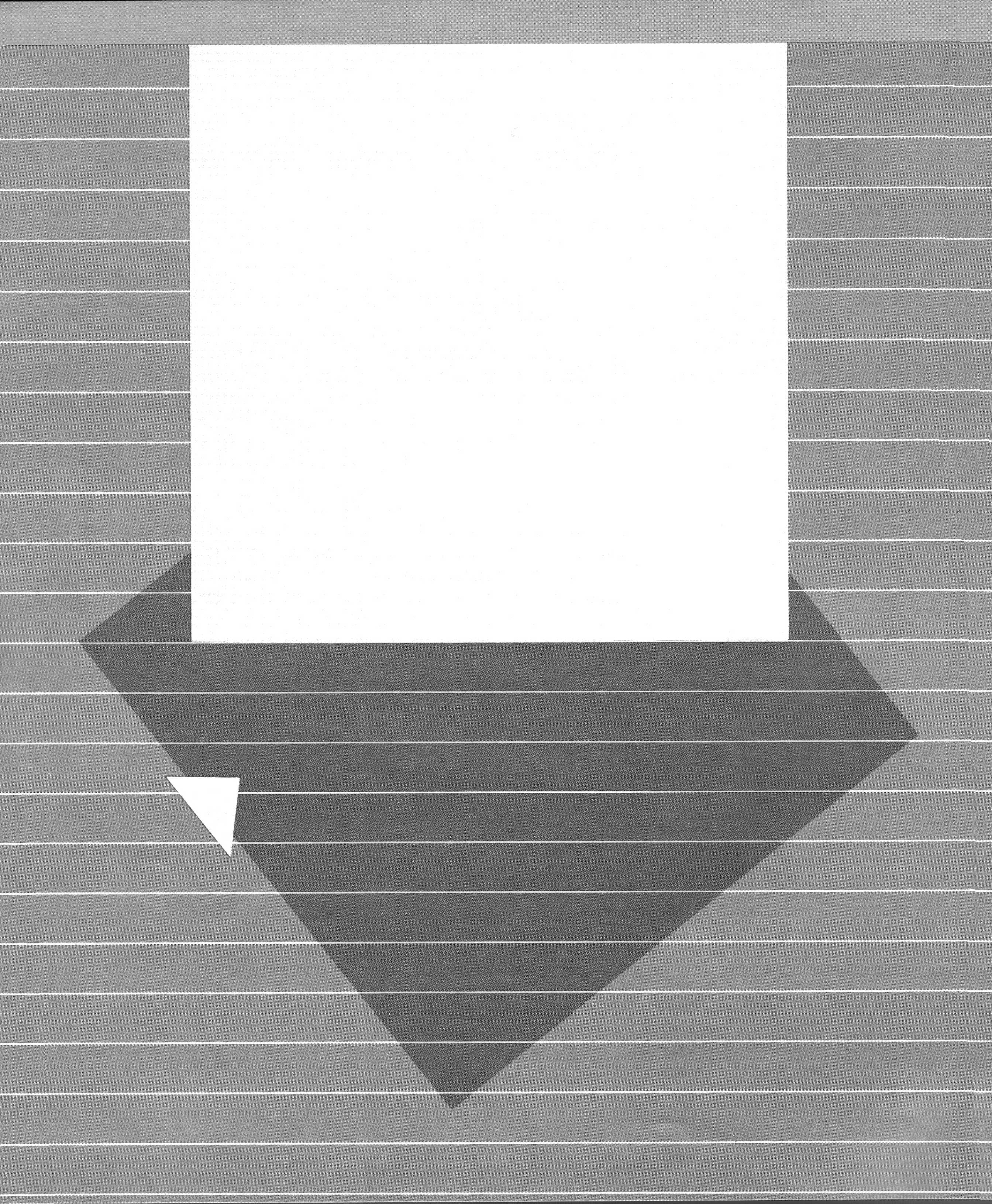
The AppleMouse II gives you an entirely new way to control your Apple II, II Plus, or IIe computer. Instead of typing commands from the keyboard, you can now interact directly with your monitor screen. You can see exactly what you are doing as you do it.

A Few Mouse Tricks

You can do things with your mouse that would be impractical with keyboard entries. For instance:

- You can create pictures directly on the screen, as if your hand were holding an artist's pencil.
- You can select menu options simply by moving the screen cursor to them, instead of having to type identifying codes.
- You can move text or graphic elements around on the screen.

All these techniques (and more) are illustrated in MousePaint, the graphics program that comes with your mouse.



Hooking It Up

To connect your mouse to your Apple computer, you must first open up your Apple and install some new parts inside it. This is easy; it takes about 20 minutes to do the whole job. This chapter covers the procedure in four sections:

1. Installing a new connector on the back panel of your Apple. There is one procedure for the Apple IIe, and a different one for the Apple II or II Plus.
2. Plugging the mouse's electronic card into your Apple.
3. Connecting the mouse.
4. Setting up the mouse's working area.

Warning
Electrical equipment can be hazardous if misused. The installation and operation of this device must be performed by an adult. Do not allow children access to the interior of your computer or let them handle its electrical cables.

Installing the New Connector

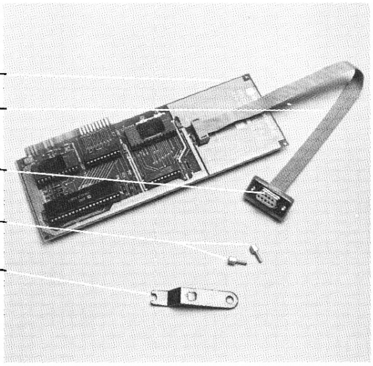
Included in your AppleMouse II package is an electronic card with a short, flat cable attached to it. You install the connector on the end of this cable in the back panel of your Apple, where it forms the new socket that you will use to connect your mouse.

Warning
Before starting any of the following procedures, turn *off* all power to your Apple system. *This is important.* You can quickly do a lot of expensive damage to your computer by working on it while its power is switched on. Leave the power cord plugged in, however, to maintain the grounding circuit to your Apple.

Apple IIe Installation

If your computer is an Apple IIe, use the connector parts shown in Figure 2-1. You will find them in a plastic bag that came in your mouse package. You do not need the other parts in the bag.

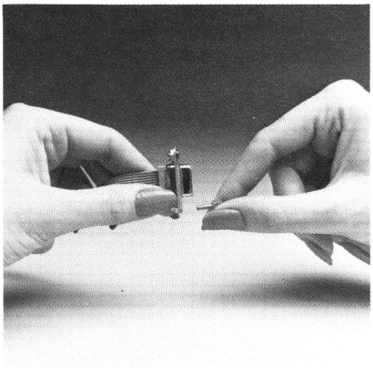
Figure 2-1. Apple IIe Connector Parts



Here is how you put these connector parts together:

1. Turn off your computer and all equipment connected to it. Do not unplug the power cord.
2. Remove the top of your Apple by lifting it up at the back. It may take some force to pop it loose. Lay the top aside.
3. Find the **panel connector** end of the internal cable. Thread one of the two jack screws into the connector, as shown in Figure 2-2. Screw it in a few turns but not tight.

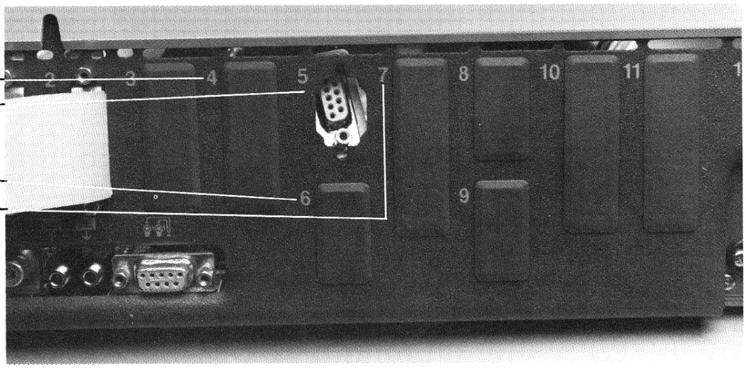
Figure 2-2. Threading a Jack Screw Into the Connector



4. Pop out the stopper from one of the four small connector holes on the back panel of your Apple. Hole number 5 (they're numbered on the back) is usually the most convenient.

5. Fit the jack screw on the internal cable connector into one of the top or bottom notches in the connector hole, as shown in Figure 2-3. The end of the connector should stick out the back of your Apple, while the cable and card remain inside.

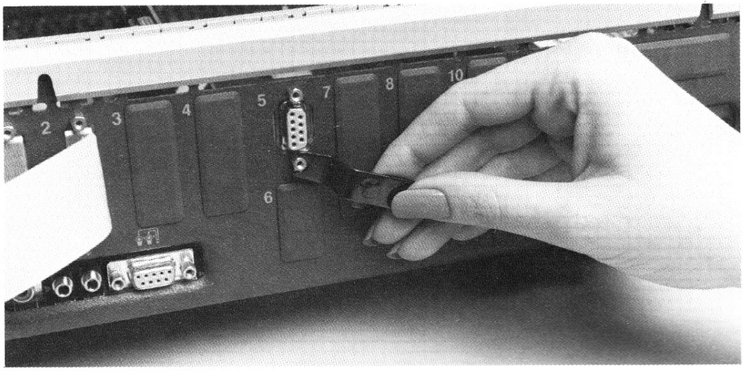
Figure 2-3. Placing the Connector in the Panel Opening (Back View)



6. Screw the other jack screw through the remaining notch in the connector hole and into the connector. Tighten both jack screws, using the Apple wrench. See Figure 2-4.

Note: Don't worry if the screws seem hard to tighten. They're made that way so they won't come loose later.

Figure 2-4. Tightening the Jack Screws

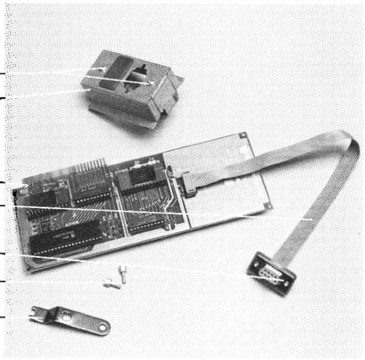


Skip the next section and go on to the section titled "Installing the AppleMouse II Card."

Apple II or Apple II Plus Installation

If your computer is an Apple II or Apple II Plus, use all of the connector parts in the plastic bag that came with your mouse. They are pictured in Figure 2-5. You will also need a small Phillips-head screwdriver.

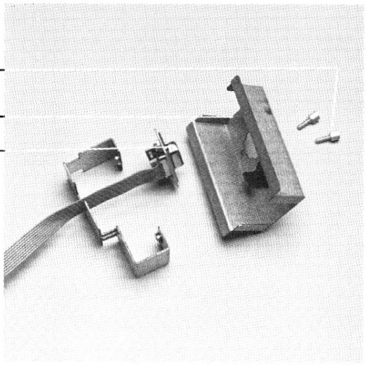
Figure 2-5. Apple II or II Plus Connector Parts


Clamp Screws
Cable Clamp
Electronic Card
Internal Cable
Panel Connector
Jack Screws
Apple Wrench

Here is how you put these connector parts together:

1. Turn off your computer and all equipment connected to it. Do not unplug the power cord.
2. Remove the top of your Apple by lifting it up at the back. It may take some force to pop it loose. Lay the top aside.
3. Disassemble the two parts of the cable clamp (included in the bag of connector parts that came with your mouse) by removing the two screws that hold them together.
4. Find the panel connector end of the cable. Thread it through the cable clamp and into one of the clamp's two openings, as shown in Figure 2-6.

Figure 2-6. Attaching the Cable to the Cable Clamp

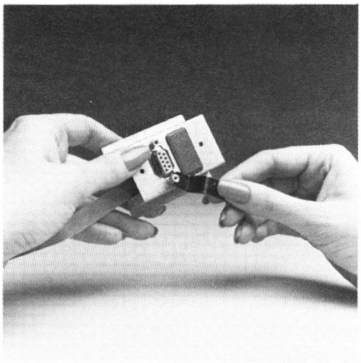

Jack Screws
Cable Clamp
Panel Connector

Note: You may have already installed a cable clamp on your Apple, if you have an Apple Silentype™ printer or other peripheral device that uses a 9-pin connector. If this is the case, you can install your AppleMouse II cable in the existing clamp's unused second connector position. Just follow this procedure, but attach the cable to the clamp that is already installed instead of putting in a new clamp.

5. Thread the two jack screws through the cable clamp and into the connector, as shown in Figure 2-7. Tighten the jack screws with the Apple wrench.

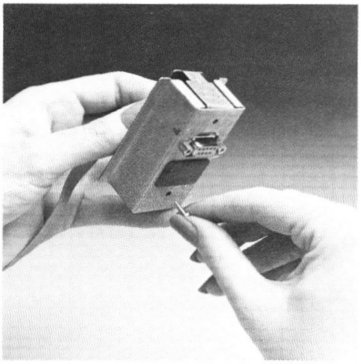
Note: Don't worry if the screws seem hard to tighten. They're made that way so they won't come loose later.

Figure 2-7. Attaching the Connector to the Clamp



6. Using the two clamp screws, assemble the two parts of the clamp loosely as shown in Figure 2-8. Don't tighten the clamp screws yet; just start them a few turns.

Figure 2-8. Assembling the Clamp



7. Notice that one end of the clamp has the word *TOP* on it. Keeping this end up, slide the clamp endways into one of the three deep notches on the back panel of your Apple, with the connector part facing outside. The middle notch is usually the most convenient to use. Slide the clamp to the bottom of the notch. Tighten the clamp screws as shown in Figure 2-9.

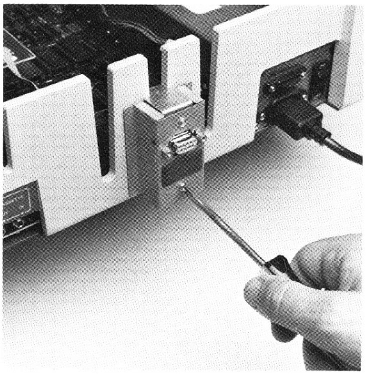


Figure 2-9. Installing the Clamp on the Back Panel

Installing the AppleMouse II Card

As with other peripheral devices, the AppleMouse II comes with an electronic card that you plug into one of the expansion slots inside your Apple.

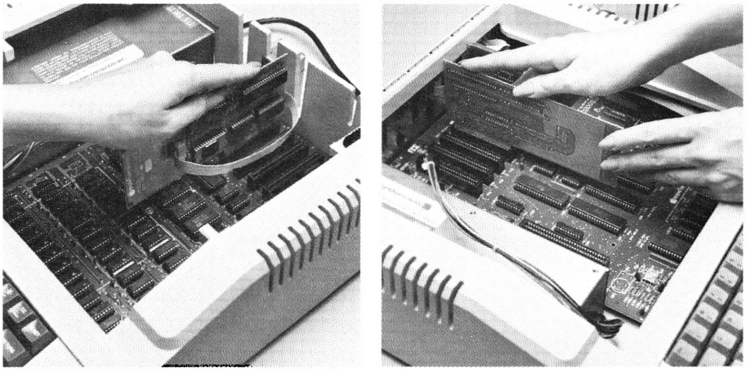
Note: You can plug the AppleMouse II Card into any expansion slot. However, we recommend that you use *slot 4.* The sample BASIC programs in Appendix A assume that the card is in slot 4.

Here's how to install the AppleMouse II Card:

1. Your Apple's power switch should still be off and the cover removed. If this is not the case, make sure that the power is **off**, and remove the cover before proceeding.
2. Place your finger momentarily on the metal cover of the power supply, on the left inside of your Apple. This removes any static electricity that you may have collected.
3. There is a row of long, slotted connector strips sticking up from the bottom of your Apple, toward the back. Each one has a number printed at its back end. Find slot 4.

4. The AppleMouse II Card has a strip of gold contact fingers, like the teeth of a comb, along one side. Plug this contact strip into slot 4 in your Apple. See Figure 2-10. Push the board down firmly, but don't force it. Try not to get fingerprints on the gold contacts.

Figure 2-10. Plugging in the Card



Apple II

Apple IIe

5. Replace the cover on your Apple by putting it in place and snapping its back edge down.

Connecting the Mouse

As a result of the foregoing installations, there is now a new connector on your Apple's back panel. This is where you hook up your AppleMouse II. Just plug the end of its tail into the new socket. Then tighten the two little screws to hold it securely.

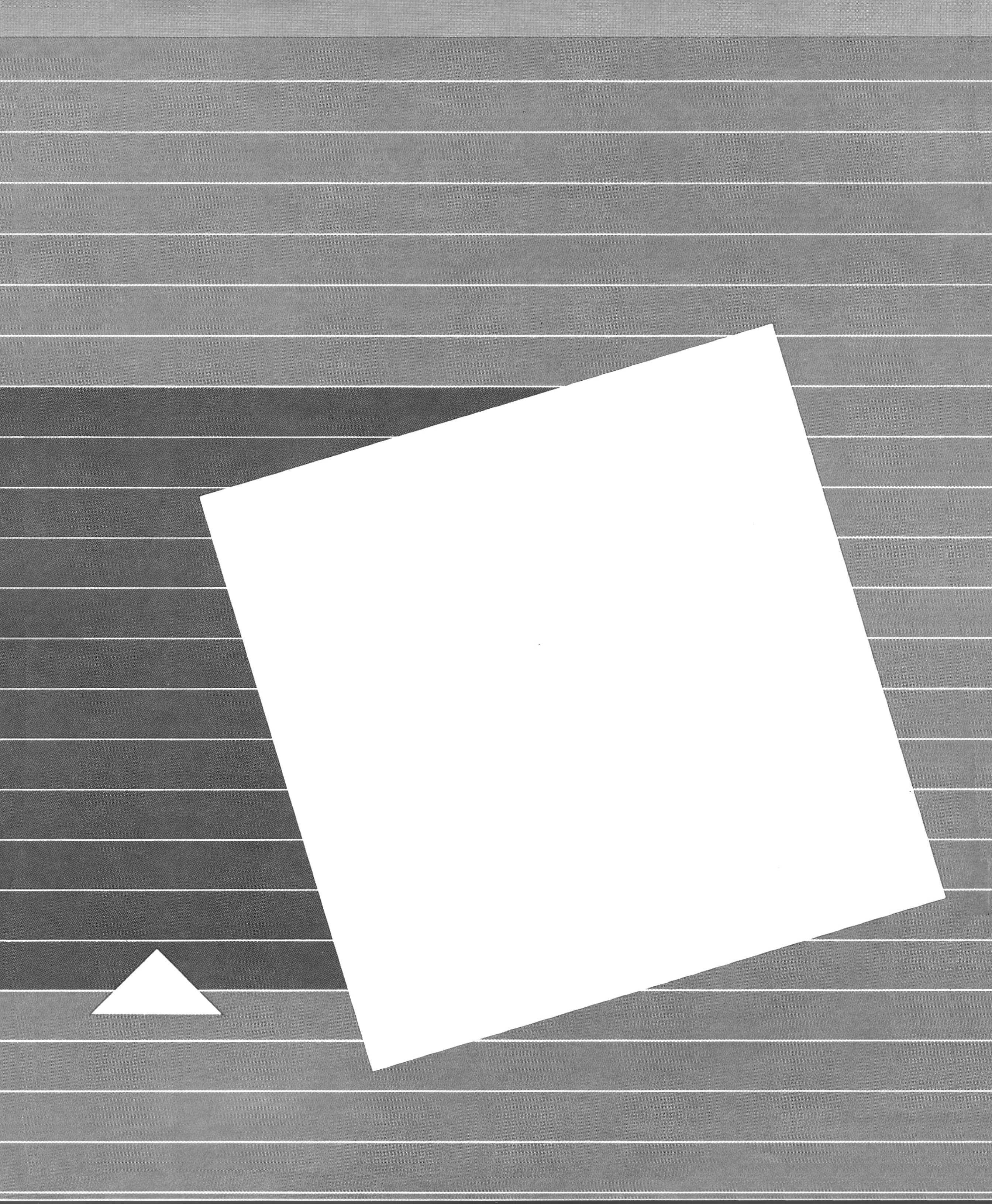
Warning
There may now be more than one mouse-type, 9-pin connector on the back of your Apple. *Make sure you plug your mouse into the correct one.* Plugging it into the wrong one (for instance, into the connector for a Silentype printer) may damage the mouse or your Apple or both. If there is more than one identical connector, label them clearly.

Your Mouse's Working Area

Because you use your mouse by moving it around on the table top, you will need to set up a working area for it. Here are some pointers.

- Your mouse's working space should be right next to your Apple. About a foot (30 cm) square of empty table top will do nicely. Make sure that it is flat and clean.
- You can move your mouse with either hand. Its 4-foot cable lets you put it on either side of your Apple.
- The rubber ball on the bottom of the AppleMouse II grips most surfaces. But occasionally it may encounter a table top that is too smooth to turn it properly. If this happens, just put down a large sheet of paper under your mouse.
- Give your mouse a clean habitat. Be careful not to let it run through wet or oily spots, dust, grit, eraser bits, cookie crumbs, or any other material that it might pick up. Such stuff can get carried inside, where it may gum up the works. For cleaning instructions, see Chapter 4.

Warning
It is possible for the small plastic pads on the underside of your mouse to mar the surface it runs on. Before operating it on highly polished furniture, lay down a paper covering to protect the finish.



MousePaint

Packed with your AppleMouse II is a program disk called *MousePaint™ Drawing Program.* (We'll call it the *MousePaint* disk for short.) It provides a fun way for you to get experience handling your mouse.

MousePaint turns your monitor screen into an artist's canvas where you can make drawings, lettering, and diagrams. You can change, move, erase, and redraw until you have created the picture you want. Then you can store it on a disk or print it on an Apple Dot Matrix Printer or ImageWriter.™

▲ **Note:** MousePaint works only with a computer of the Apple II family equipped with a minimum of 64k memory, and only when an AppleMouse II has been installed. The computer can use either a television set or a monitor screen for output.

Getting Started

Put the *MousePaint* disk in drive 1 of your Apple II computer and turn on the power. The drive will start whirring, and, after a few seconds, `Introduction to MousePaint` will appear on your screen.

Follow this self-guiding introduction carefully. It demonstrates several important techniques for using your mouse:

- How to move the **pointer** around on your screen;
- How to **click** your mouse by placing the pointer over a screen symbol and then pressing and releasing the mouse button;
- How to **drag** your mouse by moving it with its button held down;
- How to tell your computer what to do by selecting an option from a **Pull-Down menu**.

When you have finished the introduction, you will learn how to go to the main MousePaint program. The MousePaint display will appear on your screen, looking like Figure 3-1.

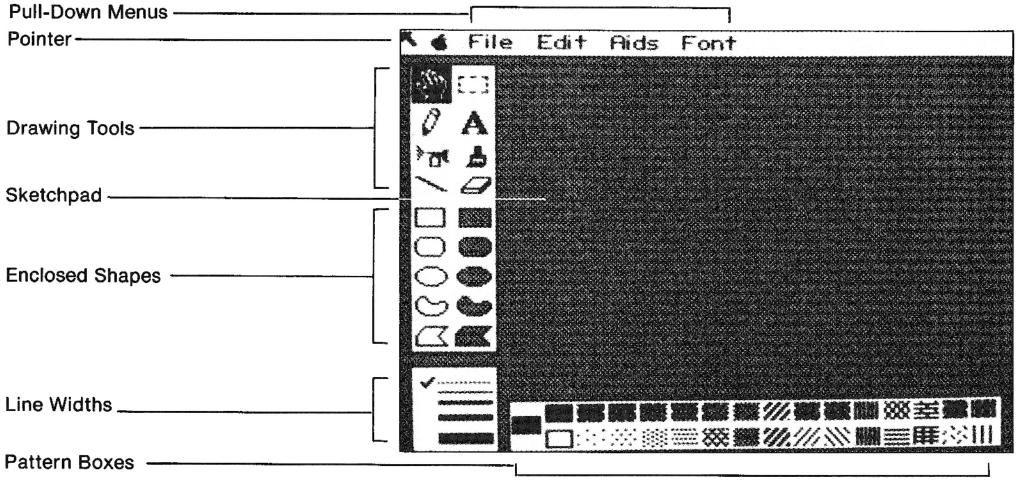


Figure 3-1. The MousePaint Display

The MousePaint Display

The parts of the MousePaint display are labeled in Figure 3-1 and are explained in detail later in this chapter. Here is a quick summary:

- The **sketchpad** is the area where you draw pictures. It starts as an empty pattern. One of the first things you will do is tell MousePaint to place a **blank screen** on it so you can start drawing.
- You use the **Pull-Down menus** to bring lists of MousePaint commands to the screen. These commands help you save your pictures on disk, cut and paste parts of them, select typestyles, and so on.
- You fetch the MousePaint **drawing tools** by means of their symbols—a pencil, a spray can, a brush, an eraser, and so on.
- The five pairs of **enclosed shapes** help you draw geometric figures. Each one has an outline form and a solid form.
- When you ask MousePaint to draw a line, you can specify its thickness by selecting one of the **line widths**.
- When you draw a solid figure, you can fill it with any one of the patterns shown in the **pattern boxes**.
- Somewhere on the display is a **pointer**. You move this around when you roll your mouse.

MousePaint's drawing tools all have names and specific functions:

- The **grabbing hand** moves your picture around so you can work on different parts of it.
- The **Editor's box** lets you mark off any part of your picture for modification by one of the Edit Menu commands.
- The **pencil** lets you sketch freehand lines.
- The **text letter** adds neat lettering to your drawings.
- The **spray can** helps you produce shading on your drawings.
- The **brush** draws a swath of black in your choice of widths.
- The **straight edge** helps you draw straight lines.
- The **eraser** rubs out unwanted parts of your pictures.

Making a Picture

Once the MousePaint display is on your screen, you are ready to create pictures. For starters, try drawing a box and writing your initials inside it. Here's how:

Getting a Blank Screen

1. Start out with the MousePaint display on your screen.
2. Roll your mouse until the pointer on the screen is on the File menu head in the top line.
3. Pull down the File menu; that is, hold down the mouse button and move the pointer downward.
4. Move the pointer to `Blank HiRes Screen`. Release the button. A blank screen will appear on the sketchpad.

Selecting Line Width and Pattern

1. Move the pointer down to the third line width from the top in the lower-left corner of the display. Click your mouse. The check mark will move to the width you selected.
2. Move the pointer to the fourth pattern box from the left in the bottom row. Click your mouse. The pattern you selected will appear in the box at the left end of the row.

Drawing a Shape

1. Move the pointer to the filled-in rectangular shape. It is halfway up the left side, just under the eraser. Click your mouse. A black box will appear around the rectangle to show that it has been selected.
2. Move the pointer onto the sketchpad. Notice that it becomes a cross. Starting from a point near the upper-left corner of the sketchpad, drag your mouse diagonally down to a spot near the lower-right corner. MousePaint will draw a large rectangle as you go. It will be filled with the pattern you selected, and bordered by the line width you selected.

Using the Brush

1. Move the pointer to the brush symbol to the left of the sketchpad. Click your mouse. The brush becomes light on dark to show that you have selected it.
2. Move the pointer onto the sketchpad and inside the patterned rectangle you just drew. Now write your initials on the pattern, using your mouse. When you hold down the button, it's like putting the brush down; when you release the button, it's like lifting the brush. Don't worry if your writing is uneven or you make some extra lines; this is just for practice.

Modifying Your Picture

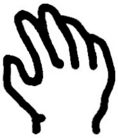
1. Move the pointer to the Editor's box in the upper-right corner of the drawing-tools section to the left of the sketchpad (this is different from the enclosed-shape box that you used before). Click your mouse. The box becomes highlighted to show you it's working.
2. Move the pointer back onto the sketchpad and drag it diagonally over the first letter of your initials. MousePaint will draw the Editor's box around that letter.
3. Pull down the Edit menu to `Invert` and release the mouse button. The part of your picture inside the Editor's box will become light on dark instead of dark on light.

If your picture looks a trifle crude, remember that even Michelangelo had to start somewhere. Now you're ready to explore MousePaint on your own. In the rest of this chapter, you'll learn about some of the things you can do with MousePaint.

Drawing Tools

The following sections describe Mousepaint's drawing tools.

Grabbing Hand

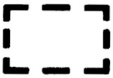


The grabbing hand lets you pull fresh paper onto your sketchpad. The space you see on your monitor screen is only part of the total sketchpad. To get more of it (or bring another part of a big picture into view), use the grabber:

1. Click the hand symbol.
2. Drag your mouse on any part of the sketchpad.

You can pull the sketchpad in any direction; MousePaint will stop you when you reach an edge. The pointer does not need to be on the picture when you drag your mouse because you are moving the underlying sketchpad, not what's drawn on it.

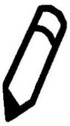
Editor's Box



The Editor's box marks off some part of your picture for further editing action. Here's how to use it:

1. Click the dotted-line box symbol.
2. Drag your mouse diagonally across the element you want to edit. MousePaint draws a dotted-line box around the element, with its opposite corners where you start and end the drag.
3. Move the pointer to `Edit` at the top of the MousePaint display. Hold down the button to see the Edit menu.
4. Move the pointer down to the Edit command you want and release the button. MousePaint will perform that function on the picture element inside the box. Editing commands are described later in this chapter.

Pencil



The pencil lets you make freehand drawings on your sketchpad:

1. Click the pencil symbol.
2. Drag your mouse on the sketchpad wherever you want to draw a line. When you release the button, you lift the pencil; when you press the button, you put it down again.

Note: The pencil writes either dark or light. Its color is the opposite of the color of the background where it first started.

Text Letter

A

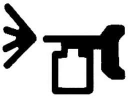
The text letter lets you add lettering to your picture. Here's how to do it:

1. Click the big *A* symbol.
2. Move the pointer to `Fonts` at the top of the MousePaint display. Hold down the button to see the menu of typestyles.
3. Move the pointer down to the typestyle you want and release the button.
4. Click your picture where you want your line of text to start.
5. Type the text you want on your keyboard.

You can move or edit lettering in your picture, just like any other element.

Caution: MousePaint uses the same memory area for lettering as for its Cut and Paste functions (see "Editing Commands" later in this chapter). When you select a lettering font, any previously cut clipping is destroyed. MousePaint will ask you if you want to destroy the clipping. Answer by pressing Y or N on your keyboard, and then pressing RETURN.

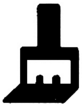
Spray Can



The spray can helps you put shading on your picture:

1. Click the sprayer symbol.
2. Drag your mouse on the sketchpad to lay down shading. When you drag over the same spot more than once, the shading gets darker.

Brush



The brush paints a swath of black in various widths. You set it up and use it like this:

1. Pull down the Aids menu and release the mouse button on `Set Brush`. A display of brush types will appear.
2. Pick the style of brushwork you want by clicking one symbol on the display.

3. Click the brush symbol in the drawing-tools section to the left of the sketchpad.
4. Drag your mouse on the sketchpad where you want the brush to paint.

Straight Edge

The straight edge draws a straight line at any angle, in various widths and patterns. Here's how you use it:

1. Click the diagonal line symbol.
2. Click one of the line widths at the bottom-left corner of the MousePaint display.
3. Click a pattern box at the bottom of the MousePaint display. The pattern you select will appear in the box at the far left. Select the plain black box for a solid line.
4. Drag your mouse on the sketchpad approximately where you want the line to be located. MousePaint will draw a neat, straight line of the width and pattern you selected between the beginning and end of the drag.

Eraser

The eraser wipes clean any part of your drawing, quickly and easily:

1. Click the eraser symbol.
2. Drag your mouse on the sketchpad wherever you want to erase.

Enclosed Shapes

Below the drawing tools, on the left side of the MousePaint display, are five pairs of shape symbols. Each has an outline form and a solid form. When you select one of the shape symbols, MousePaint draws a geometric figure on your sketchpad. If you choose the outline form, it just draws the perimeter; if you choose the solid form, it fills it with the currently selected pattern. Here's how to draw an enclosed shape:

1. Click the symbol for the kind of shape you want: rectangle, rectangle with rounded corners, oval/circle, free form, or polygon.
2. Click the line width you want for its perimeter, choosing from the selection at the lower-left corner of the MousePaint display.

3. Click a pattern box at the bottom of the MousePaint display. If you select an outline form, MousePaint will use this pattern for the border; if you select a solid form, MousePaint will fill the inside with it.
4. Depending on the shape you chose, define its size and position on your sketchpad by one of the procedures that follow.

For **rectangles**, drag your mouse along a diagonal line. MousePaint will construct a rectangle with opposite corners at the beginning and end points of the drag.

For **rounded-corner boxes**, do the same as for rectangles. The corner radius is the same for all sizes.

For **circles** and **ovals**, drag your mouse as if you were making a rectangle. MousePaint draws the largest circle or oval that will fit inside the rectangular space you have defined.

For **free-hand shapes**, drag your mouse around the outside.

For **polygons**, click your mouse at each corner.

Editing Commands

MousePaint's Editing commands let you manipulate large chunks of your picture—move them, duplicate them, delete them, or modify them. They operate only on the part of your picture that you have previously enclosed in the Editor's box. Here are the details:

Move

To move any part of your picture to another place on the sketchpad, just do this:

1. Enclose in the Editor's box the part of your picture you want to move.
2. Place the pointer inside the Editor's box and drag your mouse. The Editor's box and its contents will move with the mouse.

Caution: As long as you hold down the mouse button, you can move the Editor's box over other parts of your picture. When you release the mouse button, MousePaint erases everything covered by the box.

Cut and Paste

The Editor's Cut command scissors out the part of your picture that's in the Editor's box. The resulting clipping is not destroyed, however; MousePaint stores it in memory. When you select the Paste command, the clipping is glued back into your picture at the place you specify. Here's how to cut:

1. Click the Editor's box symbol (it's in the upper part of the drawing-tools section).
2. Drag your mouse diagonally over the part of your picture you want to cut out. MousePaint will draw the Editor's box around it.
3. Pull down the Edit menu and select Cut. The material inside the Editor's box will disappear.

To paste a clipping you have just cut into a new location, do this:

1. Pull down the Edit menu and select Paste.
2. Move the pointer (which now becomes a pointing finger) to any position on the sketchpad.
3. Click your mouse. The previously cut clipping will materialize on the sketchpad at the end of the finger.

You can paste any number of copies of one clipping in different places.

Caution: If you select two Cut commands in a row, the clipping from the first one will be destroyed. Only the most recently cut clipping is held.

Copy

The Editor's Copy command duplicates the part of your picture that's in the Editor's box, without removing the original. You use it in the same way as the Cut command, but the original remains behind on the sketchpad.

Delete

The Editor's Delete command erases everything inside the Editor's box:

1. Enclose in the Editors' box the part of your picture to be deleted.
2. Pull down the Edit menu and select `Delete`.

Did you just say "Oops"? Then use the Editor's Undo command (explained later in this chapter) to put it all back.

Invert

The Editor's Invert command changes all the white parts inside the Editor's box to black, and all the black parts to white:

1. Enclose the part of your picture to be inverted in the Editor's box.
2. Pull down the Edit menu and select `Invert`.

Flip Horizontally and Vertically

The Editor's Flip commands turn over the part of your picture inside the Editor's box, creating a mirror image. `Flip Horizontally` trades right and left; `Flip Vertically` trades top and bottom:

1. Enclose the part of your picture to be flipped in the Editor's box.
2. Pull down the Edit menu and select `Flip Horizontally` or `Flip Vertically`.

Undo

The Editor's Undo command reverses the last action you took, returning your picture to its previous state. Just pull down the Edit menu and select `Undo`. You must use this command right away, however—it works only on the very last thing you did.

Drawing Aids Commands

MousePaint's Aids menu contains several commands that help you make neat, detailed drawings.

FatBits

The FatBits command creates an enlarged view of a small part of your picture, each point of which appears as a separate black square. With FatBits, you use the MousePaint drawing tools to make finely detailed changes. You can create or erase individual points by clicking the pencil on them, fill precise areas with the brush or spray can, or edit small regions. You can also create any of MousePaint's enclosed shapes in miniature. All the changes you make will be faithfully incorporated into the original sketchpad image. Here's how to use the FatBits command:

1. Using the Move command, place part of your picture in the upper-left corner of the sketchpad.
2. Pull down the Aids menu and select FatBits.
3. Select one of the MousePaint shapes or drawing tools, as explained earlier in this chapter.
4. Follow the procedure for the shape or drawing tool you selected, working on the enlarged view.

When you have finished with FatBits, just click FatBits on the Aids menu a second time.

Show Page

The Show Page command creates a miniature image of your entire picture, including parts that are not currently in view. This lets you check your picture as a whole. It's simple:

1. Pull down the Aids menu and select `Show Page`. The miniature page will appear.
2. To move the miniature page around, place the pointer on `Show page` and drag your mouse.
3. When you are finished viewing the whole page, click your mouse on the rectangle in its upper-left corner.

Set Brush

The Set Brush command allows you to select the kind of line the brush will paint. Here's how:

1. Pull down the Aids menu and select `Set Brush`.
2. Click one of the brush styles shown on the menu.

Grid

The Grid command adjusts the positions of lines and enclosed shapes as you create them, so that they always fall on a set of invisible grid lines. This helps keep your drawing neat; you don't have to worry about positioning everything exactly right. You start and stop the Grid command by pulling down the Aids menu and clicking `Grid`. It keeps working until you tell it to stop.

Filing Commands

Once you have created the picture you want, MousePaint provides a variety of commands to help you store it or print it.

Blank Screens

MousePaint gives you a choice of blank screens to put on your sketchpad. They are listed on your sketchpad. To fetch one, just pull down the File menu and release the mouse button on your selection.

Warning
Whenever you fetch a blank screen, the picture on your previous screen is destroyed. If you want to preserve your picture in a disk file, use the Filer command Put a Copy In first.

Disk Names and Filenames

MousePaint uses the same file-naming rules as does Apple ProDOS. If you are not familiar with ProDOS, here's an abbreviated summary.

Acceptable Names

When you format a storage disk, MousePaint asks you what to name that disk. When you create a new file on a storage disk, MousePaint asks you for a filename. In either case, you must think up a name that conforms to all of the following rules:

- The name must begin with a slash followed by a letter.
- It can contain only letters, numbers, and/or periods.
- It must not be longer than 15 characters.

Warning
It is a good idea to write down your disk name and all its filenames on the disk's label. If you forget them and can't type them on your keyboard, MousePaint will be unable to fetch the corresponding files.

Calling a File

Whenever you create a new MousePaint storage file, or retrieve or delete an old one, you must type its complete name on your Apple II keyboard. The complete name consists of the disk name followed by the filename. You must use the disk *name*; a disk drive designation (such as "#4:") won't work.

Examples

Here are some examples of correct ways to designate MousePaint files:

/DISKNAME/FILENAME
/DISK.1/FILE.305
/A/A
/MY.CARTOONS/FELIX.THE.CAT

Here are some examples of disk names and filenames that won't work with MousePaint:

DISKNAME/FILENAME	(Disk name does not begin with /)
/DISK NAME/FILE NAME	(Spaces not allowed)
/DISKNAME/305.FILENAME	(/ must be followed by a letter)
/A.LONG.DISK.NAME/FILENAME	(Disk name too long).

Formatting a Storage Disk

To save your MousePaint picture, you must store it on a disk with a special format. The Format Disk command converts ordinary flexible disks into MousePaint-formatted disks.

Note: MousePaint uses the Apple ProDOS operating system. If you have the ProDOS software, you can use it to format MousePaint disks and manipulate MousePaint files.

Here's how to format a disk to accept MousePaint files:

Warning

Formatting a disk destroys all the files that were previously on it. Are you sure you want to do that? Be careful not to format your *MousePaint* disk accidentally.

1. Pull down the File menu and select Format Disk.
2. Remove the *MousePaint* disk from drive 1 of your Apple system.
3. Place the disk you want to format in drive 1, where the *MousePaint* disk was.
4. On your Apple II keyboard, type the name you want to give your storage disk. Follow the rules given earlier in this chapter in the "Disk Names and Filenames" section.
5. Press RETURN on your keyboard. Your disk drive will whirr for several seconds while it formats the disk.
6. Remove the formatted disk and replace the *MousePaint* disk.

Put a Copy In

The Put a Copy In command creates a new disk file containing a copy of the picture that's on your sketchpad. You do it this way:

1. Be sure you have a MousePaint-formatted disk in one of your disk drives.
2. Pull down the File menu and select Put a Copy In...
3. MousePaint places a new line at the top of the screen, saying Save as file named... Type the filename under which you want to store your picture (including the disk name). If you're not sure how to create a MousePaint filename, see the section "Disk Names and Filenames" earlier in this chapter.
4. Press RETURN. MousePaint opens a new disk file and stores your picture in it.

You can also use this command to duplicate a file, giving it a different name.

Get Picture

The Get Picture command fetches a picture that you previously stored on disk and places it on the MousePaint sketchpad. Here's how to retrieve a picture from your files:

1. Make sure your MousePaint-formatted storage disk is in one of your disk drives.
2. Pull down the File menu and select `Get Picture...`
3. MousePaint places a new line at the top of the screen, saying `Get picture named...` Type the name of the file that you want to retrieve (including the disk name), and then press RETURN.
4. MousePaint will fetch the file and copy it onto the sketchpad.

Warning

Whenever you get a new picture, the work presently on your sketchpad is erased.

Put Back Picture

The Put Back Picture command updates the disk file from which you originally fetched a picture by storing the current state of the MousePaint sketchpad under the same name. It thus overwrites the original file with the current picture. This command provides a handy way to save your work periodically as you are creating a picture. It's simple to do:

1. Make sure the disk that your current picture came from is still in one of your disk drives.
2. Pull down the File menu and select `Put Back Picture.` MousePaint will find the original disk file and update it.
3. Wait until the disk drive light has gone out before resuming work on your sketchpad.

Delete File

The Delete File command erases a picture stored on disk. Here's how to do it:

1. Make sure the file you want to delete is on a disk in one of your disk drives.
2. Pull down the File menu and select `Delete File...`
3. MousePaint places a new line at the top of the screen, saying `Delete file named...` Type the name of the file you want to delete, including its disk name, and press RETURN.
4. MousePaint will erase that file.

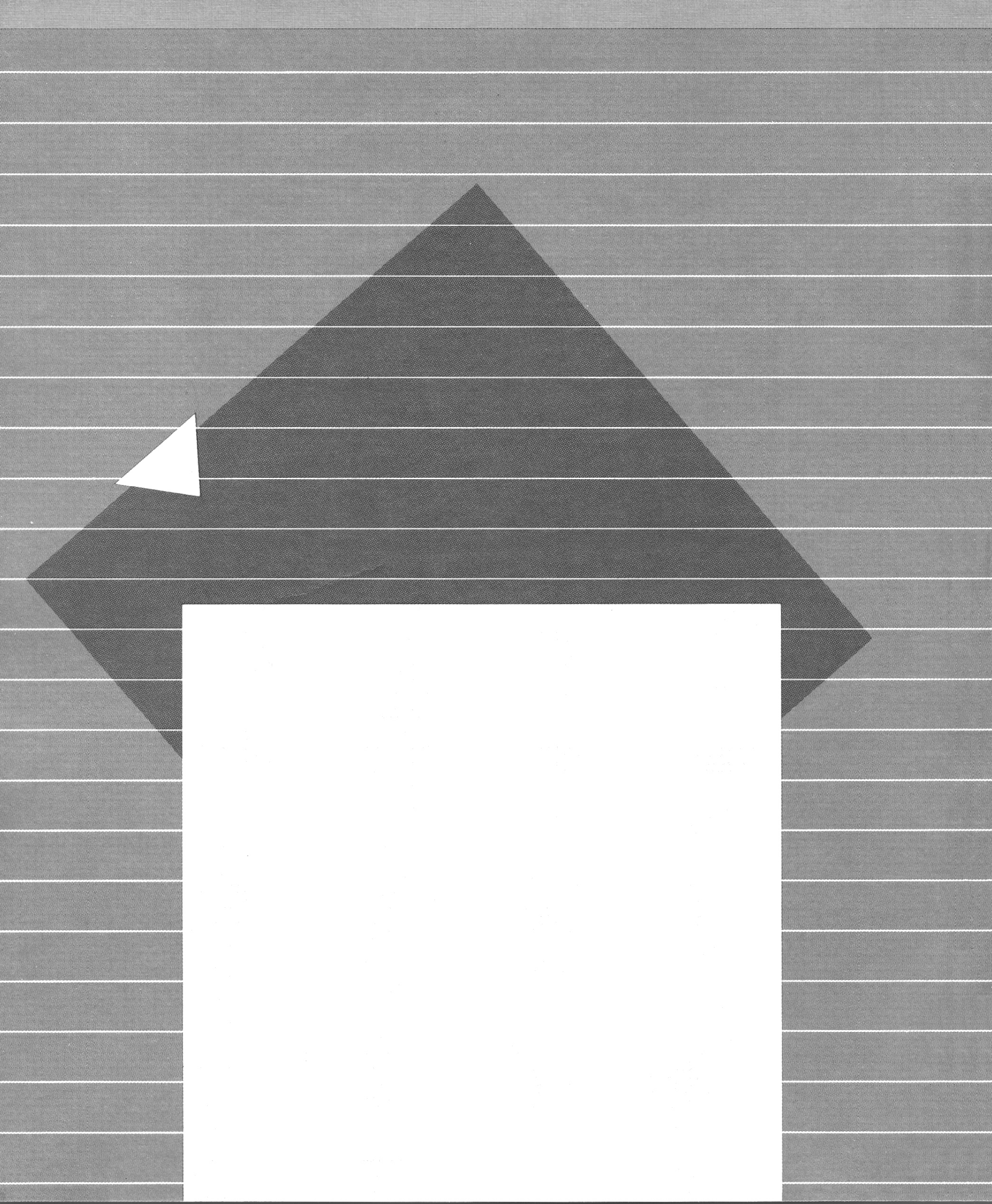
If you change your mind and want to cancel the Delete command (before completing step 3), just click your mouse.

Print Picture

The Print Picture command sends the picture on the MousePaint sketchpad to an Apple Dot Matrix Printer or Imagewriter. If you have one of these devices in your Apple system, it's easy to make a paper copy of your current artwork:

1. Make sure that your printer is turned on and has fresh paper.
2. Pull down the File menu and select `Print Picture`.
3. MousePaint places a new line at the top of the screen, asking you what slot number your printer is connected to. Type the slot number, and press RETURN.
4. MousePaint sends commands to your printer, telling it how to create a dot-by-dot copy of the image on the sketchpad.

Wait until your printer is finished before working further with MousePaint. Some pictures may take several minutes to print.

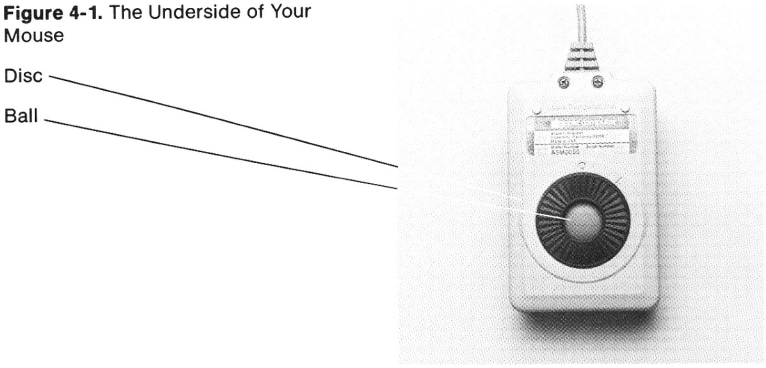


Care and Feeding

As you push your mouse about on the table, the ball on its underside may pick up dirt and carry it inside. To clean your mouse, just do the following:

1. Turn your mouse over. On its tummy you will see a black, plastic disc with a hole in its center. The disc has a round mark near its edge, which should be opposite the letter *L* (for *locked*) engraved in the mouse. See Figure 4-1.

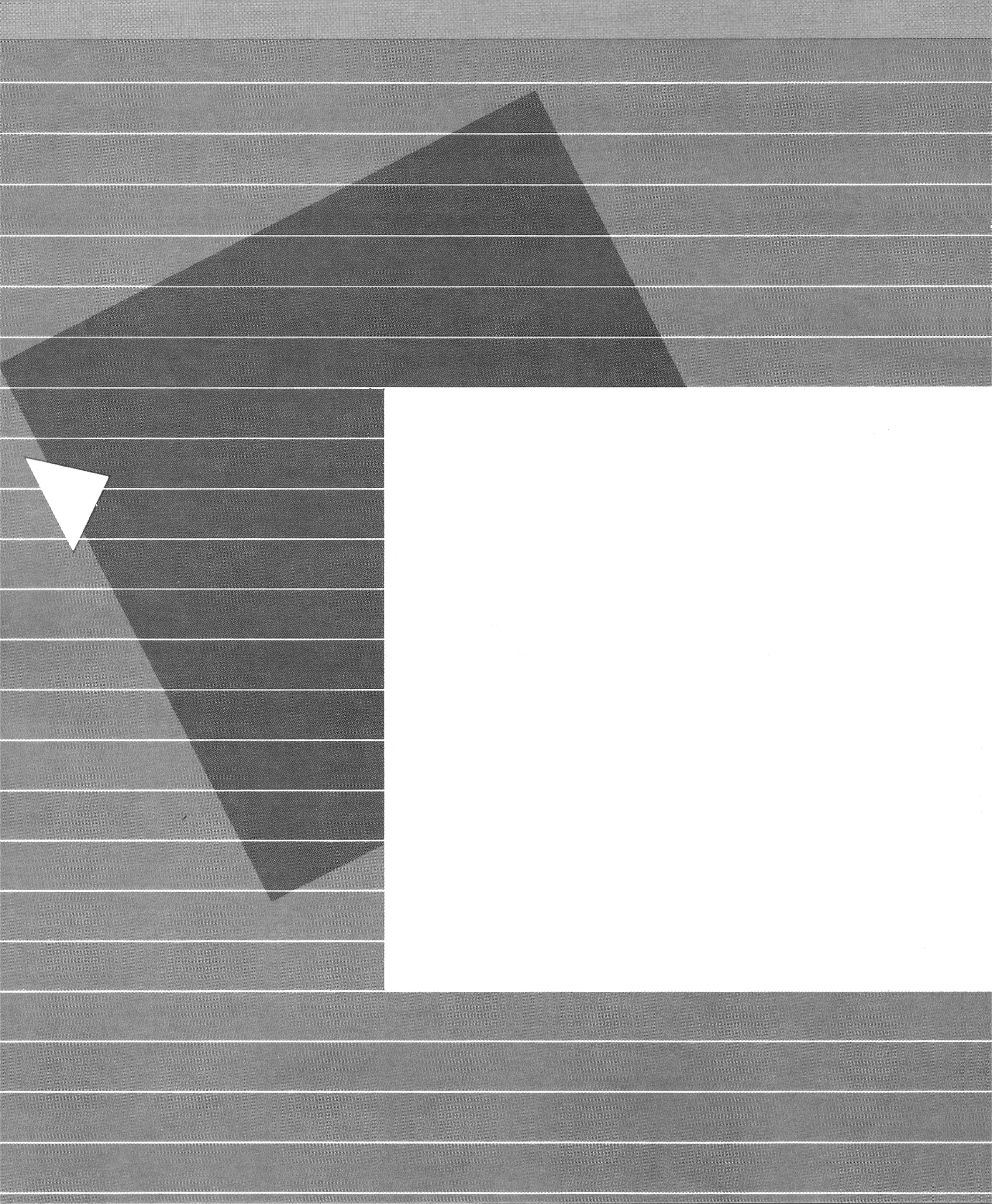
Figure 4-1. The Underside of Your Mouse



2. Turn the black disc counterclockwise one-eighth of a turn. This will bring the mark on the disc opposite the letter *O* (for *open*).
3. Cup your hands around the mouse and turn it rightside up. The black disc will fall out, and with it the ball.
4. Wipe the ball clean. If it is greasy, wash it with warm, soapy water, and dry it thoroughly with a lint-free cloth.
5. Turn the mouse back upsidedown. If there is any material in the hole on its bottom, gently pick it or shake it out. Do not try to blow it out. This may just drive the material farther inside. Do not attempt to wash out the cavity or use solvents on it.
6. Place the clean ball back into its hole in your mouse's tummy.

7. Replace the black, plastic disc, orienting it so that the spot on its rim is near the letter *O*. Turn it one-eighth of a turn clockwise, bringing the spot to the letter *L*. This should lock the disc in place.
8. Turn the mouse rightside up. It is ready for use again.

This is about all the care your AppleMouse II needs. If you keep it clean and don't drop it, it will function for many years.



Appendix A

Writing Mouse Programs in BASIC

If you understand BASIC (*Beginner's All-Purpose Symbolic Instruction Code*), you will find it easy to write programs that work with your AppleMouse II. This appendix tells you how.

The BASIC Mouse

In a BASIC program, your AppleMouse II acts like any other peripheral device—like a printer or a disk drive. You activate it with a PR# command and then extract information from it with an IN# command followed by INPUT. It returns information about its location and button status in the form of a sequence of numbers.

Calling Your Mouse

The AppleMouse II Card contains a microprocessor and a permanent program (*firmware*) that allows it to communicate with BASIC programs. So to call the mouse from a BASIC program, you must first start up the mouse's own internal program.

To do so, your BASIC program wakes up your mouse by sending the startup message `01` to its card, using the PR# command followed by a PRINT statement. If you are running Apple DOS or ProDOS, precede the PR# command with a CONTROL-D (ASCII 04) character. Assuming your mouse's card is plugged into slot 4, you can do all this with the command sequence

```
PRINT CHR$(4);"PR# 4" : PRINT CHR$(1)
```

This sends the ASCII character 01 to the device in slot 4, which places the mouse in BASIC mode and sets the mouse position numbers to zero. Don't forget to follow it by `PRINT CHR$(4);"PR# 0"` if you are subsequently going to use the screen for output.

Reading the Mouse Position

After your BASIC program has started up the mouse, it can read the mouse's position and button status (that is, whether the button has been pressed) at any time by executing an IN# 4 command followed by INPUT to three numeric variables. For instance, the sequence

```
PRINT CHR$(4);"IN# 4" : INPUT ""; A, B, C
```

places mouse position and button data into the numeric variables `A`, `B`, and `C`.

Extracting Position and Button Information

The numeric information returned by your AppleMouse II in BASIC mode has the following format:

```
A,B,+C
```

where

`A`	is the last X (horizontal) position of the mouse
`B`	is the last Y (vertical) position of the mouse
`+`	becomes − if a key on the keyboard has been pressed
`C`	indicates the status of the mouse button

`A` and `B` vary over the range 0 to +1023 (they are never negative). Assuming the mouse is oriented with its tail away from you, `A` increases as you move the mouse to the right; `B` increases as you move the mouse toward you (away from its tail).

The status digit (`C`) returns a value in the range 1 through 4, plus or minus, reporting both the present status of the mouse button and its status the last time your program executed an INPUT command:

`C`	**Current Reading**	**Last Reading**
1	Pressed	Pressed
2	Pressed	Released
3	Released	Pressed
4	Released	Released

If you press any key on your Apple keyboard, the sign of `C` changes from positive to negative. It stays negative until you reset it to positive with the command

```
POKE -16368,0
```

Turning off the Mouse

When your program is finished using the mouse, turn it off by sending it the ASCII character 00. The BASIC commands are identical to those that call the mouse (see the section "Calling Your Mouse" earlier in this appendix), except that 0 is substituted for 1 in the second PRINT statement:

```
PRINT CHR$(4);"PR# 4" : PRINT CHR$(0)
```

A Sample Program: MOUSE.MOVE

MOUSE.MOVE displays on your monitor screen the numbers (X, Y, and S) generated by the AppleMouse II. You can roll your mouse around and watch the numbers change. To exit, press any key on your Apple keyboard. Here's the listing:

```
10   HOME
20   PRINT "This is a demonstration of the Mouse"
30   PRINT CHR$(4) ; "PR#4" :  PRINT CHR$(1)
40   PRINT CHR$(4) ; "PR#0"
50   PRINT CHR$(4) ; "IN#4"
60   INPUT ""; X, Y, S
70   VTAB 10: PRINT X;"   ",Y"   ",S"   "
80   IF S > 0 THEN 60
90   PRINT CHR$(4) ; "IN#0"
100  PRINT CHR$(4) ; "PR#4": PRINT CHR$(0)
110  PRINT CHR$(4) ; "PR#0"
120  END
```

Comments

Line 10 clears the screen to black.

Line 20 prints a heading message.

Line 30 starts up the mouse's internal program.

Line 40 establishes that subsequent PRINT commands will send information to the monitor screen.

Line 50 establishes that the subsequent INPUT command will read the mouse.

Line 60 transfers mouse position and button status readings to the numeric variables X, Y, and S.

Line 70 displays the numeric variables X, Y, and S on the 10th line of the monitor screen.

Line 80 returns the program for more mouse data if no keyboard key has been pressed. If a key has been pressed, the program drops to line 90.

Line 90 re-establishes your keyboard as the input device.

Line 100 resets the mouse position data to zero.

Line 110 re-establishes the monitor screen as the output device.

Line 120 ends the program.

A Sample Program: MOUSE.DRAW

MOUSE.DRAW allows you to use your mouse to make simple drawings on your monitor screen in low-resolution graphics mode. Pressing the mouse button clears the screen so that you can start over. Pressing any key on your keyboard produces a prompting message; you can reply by pressing either RETURN to continue or ESC to terminate the program. Here's the listing:

```
10      REM  Uses mouse to draw lo-res graphics
100     GOSUB 1000 : REM  Turn on the mouse
110     PRINT CHR$(4) ; "IN#4"
120     INPUT ""; X, Y, S : REM  Read mouse data
130     IF S = 1 THEN 100 : REM  Clear the screen
140     IF S < 0 THEN 300 : REM  Time to quit?
150     REM  Scale mouse position
160     X = INT (X / 25.575)
170     Y = INT (Y / 25.575)
180     PLOT X, Y
190     GOTO 120

300     REM  Check if time to quit
310     PRINT CHR$(4) ; "IN#0"
320     VTAB 22 : PRINT "Press RETURN to cont or ESC to quit"
330     VTAB 22 : HTAB 39 : GET A$ : POKE -16368,0
340     IF A$ = CHR$(13) THEN HOME : GOTO 110
350     IF A$ <> CHR$(27) THEN 330
360     REM  Clear screen and zero mouse
370     TEXT : HOME
380     PRINT CHR$(4) ; "PR#4" : PRINT CHR$(0)
390     PRINT CHR$(4) ; "PR#0"
400     END

1000    REM  Clear screen and initialize mouse
1010    HOME : GR
1020    COLOR = 15
1030    PRINT CHR$(4) ; "PR#4" : PRINT CHR$(1)
1040    PRINT CHR$(4) ; "PR#0"
1050    RETURN
```

Comments

Line 10 reminds you what the program does.

Line 100 calls the subroutine at lines 1000 through 1050.

Line 110 establishes that the subsequent INPUT command will read the mouse.

Line 120 transfers mouse position and button status data to the numerical variables X, Y, and S.

Line 130 re-initializes the mouse if its button is pressed.

Line 140 sends the program to its exit routine if a key on the Apple keyboard has been pressed.

Line 150 reminds you of what the next two lines do.

Lines 160 and 170 convert the range of mouse position numbers (0 to 1023) to the range of low-resolution graphics coordinates (0 to 40).

Line 180 plots a point on the monitor screen.

Line 190 sends the program back for more mouse data.

Line 300 reminds you what lines 310 through 400 do.

Line 310 tells the computer to accept input from its keyboard.

Line 320 prints prompting instructions on line 22 of the screen.

Line 330 fetches your answer to the prompt and changes the button status number back to positive (it becomes negative whenever you press a key on the Apple keyboard).

Line 340 sends the program back to reporting mouse data if you pressed (RETURN).

Line 350 fetches another answer if you press any key except (ESC).

Line 360 reminds you of what happens next.

Line 370 cancels graphics mode and clears the screen.

Line 380 resets the mouse position data to zero.

Line 390 reestablishes the monitor screen as the output device.

Line 400 ends the program.

Line 1000 reminds you what the following subroutine does.

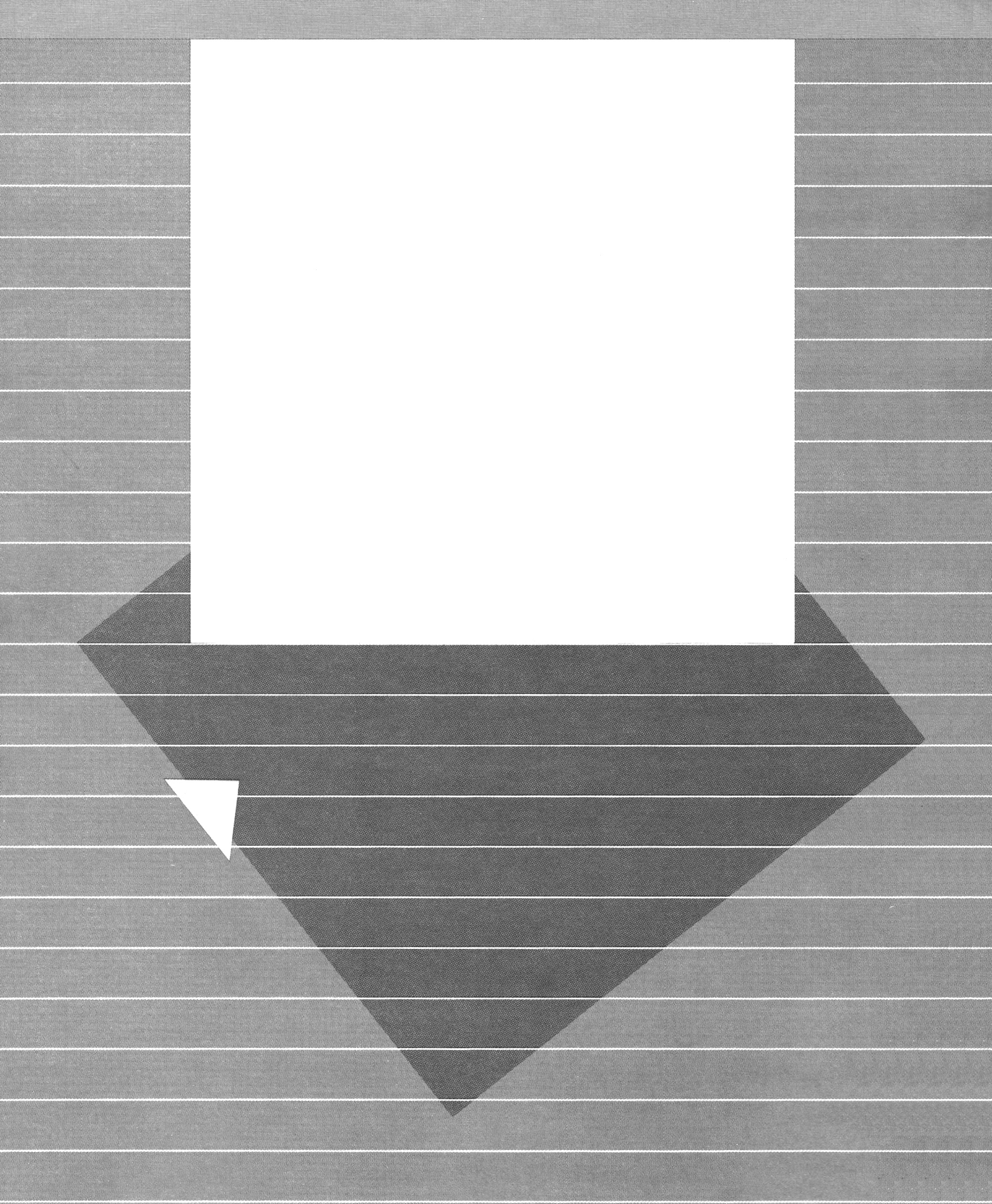
Line 1010 clears the monitor screen and sets up Apple's low-resolution graphics mode.

Line 1020 establishes that the cursor will be white.

Line 1030 starts up the mouse's internal program.

Line 1040 establishes that subsequent PRINT commands will send information to the monitor screen.

Line 1050 returns to the main program (line 110).



AppleMouse II Firmware

This appendix describes some of the 6502 assembly-language routines that are included in the firmware on the AppleMouse II circuit board. These routines allow assembly-language programs to operate the mouse.

Caution: The following description is highly technical, and is addressed to experienced 6502 programmers. The only time you need to refer to this appendix is when you are writing mouse programs in machine-level language.

Finding the Mouse

The AppleMouse II may be installed in any of the peripheral device slots 1 through 7 of the Apple II, II Plus, or IIe. Installation in slot 4 is recommended but not required. Wherever it is installed, however, it places the following signature bytes in the Apple main memory, where *n* is its slot number:

Address	Contents
$Cn0C	$20
$CnFB	$D6

Thus, your program can locate the mouse by scanning all memory locations of the forms shown here for values of *n* from 1 to 7, looking for the occurrence of both signature bytes.

Slot Numbers: In this and other memory addresses in this appendix, *n* always represents the mouse's slot number, restricted to the range 1 through 7.

Setting the Operating Mode

When power is applied to the Apple system, the mouse subsystem comes up in the *off* condition, with its position registers cleared to 0. To activate the mouse, you must load a mode byte in the 6502 accumulator and call the firmware routine SETMOUSE. Only the low-order 4 bits of the mode byte are significant:

The Operating Mode Byte

Bit 0 : Turn mouse on
1 : Enable interrupts on mouse movement
2 : Enable interrupts when button pressed
3 : Enable interrupts every screen refresh
Bits 4-7 : Reserved

Any combination of interrupts may be invoked. Mode byte values above $0F will cause SETMOUSE to return an illegal mode error.

> **Note:** The calling addresses for routines such as SETMOUSE are discussed in the section "Mouse Routines" later in this appendix.

When your program executes SETMOUSE with a mode byte of $01, it invokes **passive mode**. With any other legal mode byte, it invokes one of the **interrupt modes**. These modes are described later in this appendix.

Reading Mouse Data

Mouse position and button status data are placed in specific memory locations called **screen holes**. These are where your program looks to find out what the mouse has been doing:

Screen holes hold mouse data.

$478 + n	Low byte of X position
$4F8 + n	Low byte of Y position
$578 + n	High byte of X position
$5F8 + n	High byte of Y position
$678 + n	Reserved
$6F8 + n	Reserved
$778 + n	Button and interrupt status
$7F8 + n	Current mode

In the foregoing addresses, *n* is the mouse's slot number, which is added to the base address. For example, if the mouse lives in slot 4, the low byte of its X position will be stored at address $47C.

In the mouse's normal working position (with its tail away from the user), X increases as the mouse is moved toward the right, and Y increases as it is moved toward the user. The mouse subsystem measures X and Y movement in increments of approximately 0.020 inch (0.5 mm) over a maximum range of -32768 to +32767 (or 0 to +65535). However, both measurements are normally clamped to the range $0 to $3FF (0 to 1023 decimal). Your program can change these clamping boundaries by executing CLAMPMOUSE, as explained later in this appendix.

The button and interrupt status byte conveys the following information, where a bit set to 1 indicates true:

Interpreting Button Status

Bit 7 : Button is down
6 : Button was down at last reading
5 : X or Y changed since last reading
4 : Reserved
3 : Interrupt caused by screen refresh
2 : Interrupt caused by button press
1 : Interrupt caused by mouse movement
0 : Reserved

Passive Mode

In passive mode, the mouse does not send any interrupts to the main system. All position and button interpretation is performed on the peripheral card, and the resulting data are stored there without disturbing other routines. When you call READMOUSE, mouse information is transferred from the peripheral card to the screen holes in the main Apple memory, where your program can read it.

Passive mode represents the simplest way to manage the mouse because the operation of the mouse subsystem does not interrupt the main program. This is an important feature in applications where the mouse must function at the same time as noninterruptible peripheral devices.

Interrupt Modes

In the various interrupt modes, the mouse interrupts the main system whenever specific events occur. This allows your program to read mouse data and process the resulting information only when there is a significant change, instead of having constantly to poll the mouse. Depending on the mode byte implanted by SETMOUSE (see "Setting the Operating Mode" earlier in this appendix), the interrupt events can be any one or more of the following:

Interrupt Events

Mouse motion in any direction
Button being pressed
Screen refresh (every 1/60th second)

Upon detecting a valid interrupt event, the mouse subsystem sends an interrupt (IRQ) instruction to the Apple's 6502 processor at the end of the current monitor screen writing cycle. This allows your program to service the interrupt during the screen's vertical blanking (retrace) cycle.

If your program invokes one of the interrupt modes, it must contain an interrupt-handling routine. At a minimum, this routine must call SERVEMOUSE. SERVEMOUSE determines whether or not the interrupt was caused by the mouse, so your program can call READMOUSE if it was. Your interrupt-handling routine may also call other firmware routines, such as CLEARMOUSE, if you want.

Mouse Routines

The AppleMouse II firmware contains eight routines that your program may call. Except for SERVEMOUSE, your program must load the following values before calling any of them (*n* is the mouse's slot number):

Initializing the 6502 X and Y Registers

$Cn in the 6502 X register
$n0 in the 6502 Y register

Upon exiting any of these routines, the contents of the 6502 accumulator and X and Y registers will be undefined, except as noted in the following sections. Except for SERVEMOUSE, the status of the 6502 carry bit (C) indicates whether or not the routine was successfully executed:

C = 0	Successful execution
C = 1	Error condition

SETMOUSE

SETMOUSE starts up mouse operation according to the mode byte it finds in the 6502 accumulator (see explanation earlier). If C = 1 on exiting, the mode byte was illegal (greater than $0F). This routine does not clear any data registers or screen holes.

SERVEMOUSE

If an interrupt was caused by the mouse, SERVEMOUSE updates the status byte ($778 + n) to show which event caused the interrupt. Upon exiting, SERVEMOUSE sets C to 0 if the interrupt was caused by the mouse and sets C to 1 otherwise. SERVEMOUSE does not transfer data to the screen holes.

READMOUSE

READMOUSE transfers current mouse data to the screen holes. It sets bits 1, 2, and 3 in the status byte ($778 + n) to 0.

CLEARMOUSE

CLEARMOUSE sets the mouse's X and Y position values to $0, both on the peripheral card and in the screen holes. The button and interrupt status byte remains unchanged.

POSMOUSE

POSMOUSE sets the position registers on the peripheral card to the values it finds in the X and Y position screen holes.

Warning

Do not try to change the contents of any screen holes except the X and Y position bytes.

CLAMPMOUSE

CLAMPMOUSE establishes new value boundaries for mouse position data. The default range for X and Y position values is $0 to $3FF. CLAMPMOUSE changes the numeric limits of either the X or Y register on the mouse peripheral card, depending on the value it finds in the 6502 accumulator:

If A = 0	it changes the X coordinate limits
If A = 1	it changes the Y coordinate limits

The new boundaries are read from the Apple main memory:

$478	Low byte of lower boundary
$4F8	Low byte of higher boundary
$578	High byte of lower boundary
$5F8	High byte of higher boundary

CLAMPMOUSE destroys the contents of the mouse's X and Y position screen holes; to restore them, your program must follow it with READMOUSE.

HOMEMOUSE

HOMEMOUSE sets the mouse position registers on the peripheral card to their lower boundaries (equivalent to the upper-left corner of a screen image of the clamping window). HOMEMOUSE does not update the screen holes; to change the screen-hole values to home position, your program must follow it with READMOUSE.

INITMOUSE

INITMOUSE sets the internal default values for the mouse subsystem and synchronizes it with the monitor screen's vertical blanking cycle. Your program must call INITMOUSE before any other mouse routines. A typical sequence of calls to initialize a mouse application program would be

INITMOUSE
SETMOUSE
CLEARMOUSE

With the Apple II and Apple II Plus, INITMOUSE overwrites page 1 of the graphics screen memory; hence, your program should call it before creating any screen displays.

Calling the Mouse Routines

The mouse firmware contains a table that gives you the low bytes of the entry addresses of its routines. The high byte is always $Cn, where *n* is the mouse's slot number. This table occupies addresses $Cn12 through $Cn19:

$Cn12	Low byte of SETMOUSE entry point address
$Cn13	Low byte of SERVEMOUSE entry point address
$Cn14	Low byte of READMOUSE entry point address
$Cn15	Low byte of CLEARMOUSE entry point address
$Cn16	Low byte of POSMOUSE entry point address
$Cn17	Low byte of CLAMPMOUSE entry point address
$Cn18	Low byte of HOMEMOUSE entry point address
$Cn19	Low byte of INITMOUSE entry point address

Thus, for example, if the mouse lives in slot 4, the entry point for the routine SETMOUSE can be calculated by adding $C400 to the contents of address $C412. Your program can use these values as the basis for constructing a jump table (an array of values used for JMP instructions with indirect addressing), from which it can call all the mouse routines.

A Sample Program

The following assembly-language program illustrates some of the points discussed in this appendix. It runs in an Apple ProDOS environment, displaying mouse information on the monitor screen. To minimize program length, we have taken some shortcuts; in particular, we have used self-modifying code (lines 104 and 125-128) to call the mouse firmware routines, instead of constructing a jump table.

```
0000:             1 *      APPLEMOUSE II SAMPLE PROGRAM

1000:      1000   3        ORG   $1000

1000:             5 ****************************************************
1000:             6 *                                                  *
1000:             7 * AppleMouse II sample program                     *
1000:             8 *    1. Tests for an AppleMouse card in the system; *
1000:             9 *         if none an error message is displayed.   *
1000:            10 *    2. Turns the Mouse on in passive mode.        *
1000:            11 *    3. Does continual READMOUSEs and displays     *
1000:            12 *         the data in the following form:          *
1000:            13 *                                                  *
1000:            14 *      X=xxxx   Y=yyyy   B=bb   x,y,b are in hex   *
1000:            15 *                                                  *
1000:            16 *    4. When a key on the Apple is pressed the     *
1000:            17 *         mode is set to 0 (mouse is turned off).  *
1000:            18 *                                                  *
1000:            19 *                                                  *
1000:            20 ****************************************************

1000:            22            MSB   ON
1000:      F941  23 PRNTAX     EQU   $F941      ;A=1st byte - X=2nd byte
1000:      0006  24 TMP        EQU   $6         ;Zero page tmps are $6/$7
1000:      0022  25 TOP        EQU   $22        ;Top of screen variable
1000:      0024  26 CH         EQU   $24        ;Cursor horizontal position
1000:      0025  27 CV         EQU   $25        ;Cursor vertical position
1000:      03D0  28 WARM       EQU   $3D0       ;Restart ProDOS
1000:      FC22  29 VTAB       EQU   $FC22      ;Moves cursor to CH/CV
1000:      C000  30 KEY        EQU   $C000      ;- if key pressed
1000:      C010  31 KEYSTROBE  EQU   $C010      ;Clears keystrobe
1000:      FD8E  32 CROUT      EQU   $FD8E      ;Output a carriage return
1000:      FDDA  33 PRBYTE     EQU   $FDDA      ;Prints A reg in Hex
1000:      F948  34 PRBLNK     EQU   $F948      ;Prints 3 blank spaces
1000:      FD0C  35 RDKEY      EQU   $FD0C      ;Wait for a keypress
1000:      FDED  36 COUT       EQU   $FDED      ;Output 1 char to screen
1000:      FB39  37 TEXT       EQU   $FB39      ;Set screen for text
1000:      FC58  38 HOME       EQU   $FC58      ;Clear screen
```

```
1000:        03B8   40 XL        EQU   $3B8       ;+Cn=Low byte of abs X pos
1000:        0438   41 YL        EQU   $438       ;+Cn=Low byte of abs Y pos
1000:        04B8   42 XH        EQU   $4B8       ;+Cn=High byte of abs X pos
1000:        0538   43 YH        EQU   $538       ;+Cn=High byte of abs Y pos
1000:        06B8   44 BUTTON    EQU   $6B8       ;+Cn=Button status
1000:               45 *                       Bit 7 = current status
1000:               46 *                       Bit 6 = status on last READMOUSE

1000:               48 *  Table of offsets to mouse entry points
1000:               49 *
1000:        0012   50 SETMOUSE   EQU  $12
1000:        0013   51 SERVEMOUSE EQU  $13
1000:        0014   52 READMOUSE  EQU  $14
1000:        0015   53 CLEARMOUSE EQU  $15
1000:        0016   54 POSMOUSE   EQU  $16
1000:        0019   55 INITMOUSE  EQU  $19
1000:        0017   56 CLAMPMOUSE EQU  $17
1000:        0018   57 HOMEMOUSE  EQU  $18

1000:               59 *****************************************************
1000:               60 *  MAIN routine                                     *
1000:               61 *                                                   *
1000:               62 *****************************************************
1000:        1000   63 BEGIN     EQU   *
1000:20 5A 10       64           JSR   SETUP      ;Clear screen, type header
1003:20 7D 10       65           JSR   CHECKSLOTS ;Which slot has mouse?

1006:               67 *  Call INITMOUSE to start mouse card
1006:A0 19          68           LDY   #INITMOUSE ;Pick up offset to offset
1008:20 46 10       69           JSR   CALLCARD   ;Call the Mouse card

100B:               71 *  Call SETMOUSE to turn card on
100B:A0 12          72           LDY   #SETMOUSE  ;Pick up offset to offset
100D:A9 01          73           LDA   #$01       ;Set passive mode
100F:20 46 10       74           JSR   CALLCARD   ;Call the Mouse card
1012:2C 10 C0       75           BIT   KEYSTROBE  ;Kill any keys hit

1015:        1015   77 MAINLOOP  EQU   *
1015:A0 14          78           LDY   #READMOUSE ;Pick up offset to offset
1017:A9 88          79           LDA   #$88       ;Dont care about A
1019:20 46 10       80           JSR   CALLCARD   ;Call the Mouse card

101C:20 C4 10       82           JSR   PRINTLINE  ;Print current mouse data

101F:2C 00 C0       84           BIT   KEY        ;Test for stop
1022:10 F1   1015   85           BPL   MAINLOOP   ;No keypress: Read again
```

```
1024:2C 10 C0        ALLDONE  BIT   KEYSTROBE  ;Kill key hit
1027:A0 12                    LDY   #SETMOUSE  ;Pick up offset to offset
1029:A9 00                    LDA   #$00       ;Set off mode
102B:20 46 10                 JSR   CALLCARD   ;Call the Mouse card
102E:20 39 FB                 JSR   TEXT       ;Set screen to normal
1031:A0 00                    LDY   #$00       ;Show restart message
1033:         1033   RSTMSGLP EQU   *
1033:B9 81 11                 LDA   RSTMSG,Y   ;Pick up char
1036:F0 06    103E            BEQ   FINALLY
1038:20 ED FD                 JSR   COUT       ;Show it
103B:C8                       INY
103C:D0 F5    1033            BNE   RSTMSGLP
103E:         103E   FINALLY  EQU   *
103E:4C D0 03                 JMP   WARM       ;Restart ProDOS

1041:         0001   CN       DS    1,0        ;Cn is stored here
1042:         0001   N0       DS    1,0        ;n0 is stored here
1043:4C 00 00        TOCARD   JMP   $0000      ;Operand modified by prgm

1046:                ***********************************************************
1046:                *  Subroutines                                            *
1046:                *                                                         *
1046:                ***********************************************************

1046:                ***********************************************************
1046:                *  CALLCARD                                               *
1046:                *      Routine which sets up registers for indirect       *
1046:                *        jump to the Mouse firmware                       *
1046:                *      User loads offset to offset in Y reg               *
1046:                *        A contains byte to be passed to firmware         *
1046:                *                                                         *
1046:                ***********************************************************

1046:         1046   CALLCARD EQU   *
1046:48                       PHA              ;Save user information
1047:B1 06                    LDA   (TMP),Y    ;Find offset
1049:AE 41 10                 LDX   CN         ;Set up registers for ...
104C:AC 42 10                 LDY   N0         ; ... what firmware expects
104F:8D 44 10                 STA   TOCARD+1   ;Store for ...
1052:8E 45 10                 STX   TOCARD+2   ; ...indirect jump
1055:68                       PLA
1056:20 43 10                 JSR   TOCARD     ;Jump indirect to card
1059:60                       RTS
```

```
105A:              132 *******************************************************
105A:              133 *  SETUP                                              *
105A:              134 *      Clears text screen, prints header              *
105A:              135 *                                                     *
105A:              136 *******************************************************
105A:        105A 137 SETUP     EQU   *
105A:20 39 FB      138           JSR   TEXT       ;Set screen to text
105D:20 58 FC      139           JSR   HOME       ;Clear screen
1060:A0 00         140           LDY   #$00       ;Put up message on screen
1062:        1062 141 HDRLOOP   EQU   *
1062:B9 3E 11      142           LDA   HEADER,Y
1065:F0 06   106D 143           BEQ   BEGIN1     ;0 is terminator
1067:20 ED FD      144           JSR   COUT       ;Output it to screen
106A:C8            145           INY
106B:D0 F5   1062 146           BNE   HDRLOOP    ;Go for more
106D:        106D 147 BEGIN1    EQU   *
106D:A9 03         148           LDA   #$03       ;Set top of screen
106F:85 22         149           STA   TOP
1071:A9 00         150           LDA   #$00       ;Tab down and set new top
1073:85 24         151           STA   CH
1075:A9 03         152           LDA   #$03
1077:85 25         153           STA   CV
1079:20 22 FC      154           JSR   VTAB
107C:60            155           RTS
```

```
107D:                ***************************************************
107D:                *                                                 *
107D:                *  CHECKSLOTS                                     *
107D:                *       Routine to check the 7 peripheral slots   *
107D:                *         for the signature bytes of the Mouse card *
107D:                *         $Cn0C = $20                             *
107D:                *         $CnFB = $D6                             *
107D:                *       Returns values for $Cn and $n0            *
107D:                *         in user defined locations CN and N0     *
107D:                *                                                 *
107D:                ***************************************************
107D:          107D  CHECKSLOTS EQU  *
107D:A2 07                      LDX  #$07       ;Test 7 slots only
107F:A9 00                      LDA  #$00       ;Lo byte is zip
1081:85 06                      STA  TMP
1083:A9 C8                      LDA  #$C8       ;Max slot+1
1085:85 07                      STA  TMP+1
1087:          1087  LOOP1      EQU  *
1087:C6 07                      DEC  TMP+1      ;Decrement Cn
1089:CA                         DEX             ;Decrement # to do
108A:30 21     10AD             BMI  EXIT       ;Error exit if here
108C:A0 0C                      LDY  #$0C       ;1st ID byte is here
108E:B1 06                      LDA  (TMP),Y    ;What do we have?
1090:C9 20                      CMP  #$20       ;1st ID byte is $20
1092:D0 F3     1087             BNE  LOOP1      ;If no match try next card
1094:A0 FB                      LDY  #$FB       ;Last ID byte is here
1096:B1 06                      LDA  (TMP),Y    ;What do we have?
1098:C9 D6                      CMP  #$D6       ;Must be a 'V' to match
109A:D0 EB     1087             BNE  LOOP1      ;If no match try next card
109C:A5 07                      LDA  TMP+1      ;If match fall to here
109E:8D 45 10                   STA  TOCARD+2   ;Save for later
10A1:8D 41 10                   STA  CN
10A4:0A                         ASL             ;Make $n0 for Y
10A5:0A                         ASL
10A6:0A                         ASL
10A7:0A                         ASL
10A8:8D 42 10                   STA  N0         ;Save for later
10AB:D0 16     10C3             BNE  START      ;Go start program

10AD:          10AD  EXIT       EQU  *          ;Error loop exit
10AD:20 39 FB                   JSR  TEXT       ;Set text mode
10B0:20 58 FC                   JSR  HOME       ;Clear the screen
10B3:A0 00                      LDY  #$00       ;Set index to 00
10B5:          10B5  ERRLOOP    EQU  *
10B5:B9 11 11                   LDA  NOCARDMSG,Y
10B8:F0 06     10C0             BEQ  WARMSTART
10BA:20 ED FD                   JSR  COUT
10BD:C8                         INY             ;Go to next char to show
10BE:D0 F5     10B5             BNE  ERRLOOP
10C0:          10C0  WARMSTART  EQU  *
10C0:4C D0 03                   JMP  WARM       ;Restart ProDOS
10C3:60              START      RTS
```

```
10C4:              ********************************************************
10C4:              *    PRINTLINE                                          *
10C4:              *       Prints a line of Mouse data                     *
10C4:              *                                                       *
10C4:              ********************************************************
10C4:         10C4 PRINTLINE EQU  *
10C4:A9 D8                   LDA  #'X
10C6:20 ED FD                JSR  COUT       ;Print an X
10C9:A9 BD                   LDA  #'=
10CB:20 ED FD                JSR  COUT       ;Print '='
10CE:AE 45 10                LDX  TOCARD+2   ;Pick up $Cn
10D1:BC B8 04                LDY  XH,X
10D4:BD B8 03                LDA  XL,X
10D7:AA                      TAX
10D8:98                      TYA             ;Set high value in A
10D9:20 41 F9                JSR  PRNTAX     ;Print X value
10DC:20 48 F9                JSR  PRBLNK     ;Print 3 blanks
10DF:A9 D9                   LDA  #'Y
10E1:20 ED FD                JSR  COUT       ;Print a Y
10E4:A9 BD                   LDA  #'=
10E6:20 ED FD                JSR  COUT       ;Print '='
10E9:AE 45 10                LDX  TOCARD+2   ;Pick up $Cn
10EC:BC 38 05                LDY  YH,X
10EF:BD 38 04                LDA  YL,X
10F2:AA                      TAX
10F3:98                      TYA             ;Put high byte in A
10F4:20 41 F9                JSR  PRNTAX     ;Print Y value
10F7:20 48 F9                JSR  PRBLNK     ;Print 3 blanks
10FA:A9 C2                   LDA  #'B
10FC:20 ED FD                JSR  COUT       ;Print a B
10FF:A9 BD                   LDA  #'=
1101:20 ED FD                JSR  COUT       ;Print '='
1104:AE 45 10                LDX  TOCARD+2   ;Pick up $Cn
1107:BD B8 06                LDA  BUTTON,X
110A:20 DA FD                JSR  PRBYTE     ;Print button status
110D:20 8E FD                JSR  CROUT      ;Go to next line
1110:60                      RTS             ;All done with printing
```

```
1111:              249 ******************************************************
1111:              250 *  Messages                                          *
1111:              251 *                                                    *
1111:              252 ******************************************************

1111:        1111  254 NOCARDMSG EQU   *
1111:8D 8D         255           DFB   $8D,$8D
1113:D4 C8 C5 D2   256           ASC "THERE IS NO APPLEMOUSE CARD INSTALLED!"
1139:8D 87 87 87   257           DFB   $8D,$87,$87,$87,$00

113E:        113E  259 HEADER    EQU   *
113E:AA AA AA AA   260           ASC "***** APPLEMOUSE TEST ROUTINE *****"
1161:8D            261           DFB   $8D
1162:A0 A0 A0 A0   262           ASC "           PRESS ANY KEY TO STOP."
117F:8D 00         263           DFB   $8D,$00

1181:        1181  264 RSTMSG    EQU   *
1181:8D            265           DFB   $8D
1182:D4 D9 D0 C5   266           ASC "TYPE CALL 4096 AND RETURN TO RESTART."
11A7:8D 00         267           DFB   $8D,$00

?1024 ALLDONE        106D BEGIN1        ?1000 BEGIN          06B8 BUTTON
 1046 CALLCARD       107D CHECKSLOTS       24 CH           ?   17 CLAMPMOUSE
?  15 CLEARMOUSE     1041 CN             FDED COUT           FD8E CROUT
   25 CV             10B5 ERRLOOP        10AD EXIT           103E FINALLY
 1062 HDRLOOP        113E HEADER        ?  18 HOMEMOUSE      FC58 HOME
   19 INITMOUSE      C010 KEYSTROBE      C000 KEY            1087 LOOP1
 1015 MAINLOOP       1042 N0             1111 NOCARDMSG     ?  16 POSMOUSE
 F948 PRBLNK         FDDA PRBYTE         10C4 PRINTLINE      F941 PRNTAX
?FD0C RDKEY            14 READMOUSE      1181 RSTMSG         1033 RSTMSGLP
?  13 SERVEMOUSE        12 SETMOUSE       105A SETUP          10C3 START
 FB39 TEXT              06 TMP            1043 TOCARD           22 TOP
 FC22 VTAB           10C0 WARMSTART      03D0 WARM           04B8 XH
 03B8 XL             0538 YH             0438 YL
```

Downloaded from www.Apple2Online.com

apple computer®

20525 Mariani Avenue
Cupertino, California 95014
(408) 996-1010
TLX 171-576